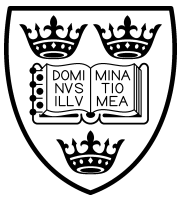
Credit Networks and Agent Games



David Buttle

Oriel College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2004

This thesis is dedicated to my mum and dad - for their love and support.

Acknowledgements

I would firstly like to thank my supervisors; Jeff Dewynne, Damion Challet, and Sam Howison - without their direction, ideas and support this work would not have been possible. A big thank you to Roman Walczak and Carole Jordan for making me believe I was up to the task. I would also like to thank Dave Smith and Tino Kluge for our many useful discussions, some of which were even on maths and finance!! I would like to thank IBM Informix for sponsoring my work over the three years and the whole of Ociam for being a supportive and fun department in which to study. Finally I would like to thank my family and friends. Thanks to all of you this has been one of the most rewarding and enjoyable periods of my life.

Abstract

This thesis is divided into three parts; an intensity based network model of firm default, an agent based network model of firm default, and an agent based model of feedback effects from dynamic hedging. The common theme among all three parts is the application of ideas from both physics and mathematics to the solution of problems motivated by the financial markets.

Less broadly, in the first two parts, the common themes are credit markets, networks, and dependent defaults. Part one tackles the problem of default dependence from a probabilistic perspective, modeling the default of companies as generalised Poisson processes, with the default dependence structure given by a network. We present a mathematical framework to solve a generalised version of the Jarrow Yu model of looping defaults [27] and study the relationship between network structure and the resilience of a network of firms to default events. Using this model we then show how to price simple multi-name credit products such as kth to default baskets.

Part two again considers dependent defaults, but here the network is dynamic and firms are modelled as simple agents, defined by strategies, whose interactions determine a network of trading links. Using our agent-based network model of firm default we study network structure and their degree distributions, firm lifetimes, and look for evidence of agent learning and default clustering. We then study the effect of default on a network of firms and the response of remaining firms to that default event.

Part three considers a relatively more established agent based framework, called the Minority Game. We first describe in detail the Minority Game and discuss its suitability as a market model. We then show how it may be applied to modelling the actions of traders delta hedging a short option position. We show that for a variety of option positions, in a sufficiently illiquid market feedback effects arise from the actions of the traders as their trades impact upon the underlying market.

Contents

Chapter 1

Preface to Credit Networks and Agent Games

The market for credit derivatives has grown rapidly over recent years. The total notional of all credit derivatives is now estimated to be around 5 trillion dollars. A large part of the market is made up of portfolio based credit products, whose value is determined by the default dependency between the underlying assets. Understanding the nature of this dependency and how best to model it is currently of great interest to both academics and financial practitioners alike, and up to now a wide variety of approaches have been developed.

Similarly, in econophysics and beyond, the use and study of both networks and agent-based models has also seen significant growth over recent years. And very recently, we are beginning to see exciting new research which combines innovation in both these areas under the broad heading of complex agent-based dynamic networks.

This thesis is divided into three parts; an intensity based network model of firm default, an agent based network model of firm default, and an agent based model of feedback effects from dynamic hedging. The common theme among all three parts is the application of ideas from both physics and mathematics to the solution of problems motivated by the financial markets. Each part naturally leads on to the next and, while they may all be taken as individual studies in their own right, a better picture in gleaned by reading the thesis through as a whole.

Less broadly, in the first two parts, the common themes are credit markets, networks, and dependent defaults. Part one tackles the problem of default dependence from a probabilistic perspective, modeling the default of companies as generalised Poisson processes, with the default dependence structure given by a network. Part two again considers dependent defaults, but rather than consider a static network,

here the network is dynamic. Firms are modelled as simple agents, defined by strategies, whose interactions determine a network of trading links. The common theme between parts two and three is the use of agent based simulations to model financial systems. Whereas part two considers a novel agent model designed specifically to model credit default, part three considers a relatively more established agent based framework, called the Minority Game, and studies how this might be applied to modeling the actions of traders delta hedging a short option position.

Therefore, in reference to the title of the thesis, *credit networks* relates to parts one and two, and *agent games* to parts two and three, with part two linking what may initially appear two fairly unrelated topics.

Part I

An Intensity Based Network Model of Firm Default

1.1 Overview

In this part of the thesis we study an intensity based approach to modeling the default dependence between firms using networks. Firstly, we introduce briefly some of the existing work that has taken place in modeling default dependence. We then detail how networks have already been applied to the study of financial markets and introduce some of the most recent areas of study in complex network analysis including small-world networks.

We then describe in detail the set up of our model and show how the model is a development of one proposed by Jarrow and Yu [27]. We develop a mathematical framework to solve for the expected time to default for any network structure and then study the effect network structure plays on the resilience of a network to default events. We show that it is possible to measure the stability of a network using simple network measures used in complex network analysis. We then show how simple multi-name credit products can be priced using the model.

Finally, we review the success of the model and assess any further modifications and applications that may provide interesting results.

Chapter 2

Introduction to the Intensity Based Network Model of Firm Default

2.1 Credit Markets and Current Credit Modeling

In this section we review the current state of the credit market and describe some of the most prevalent default based products being traded today. We then give a very brief overview of the structural and reduced form approaches to modeling default, and then describe in more detail current work in the study of default contagion and infectious defaults.

2.1.1 The Credit Derivative Market

The credit derivative market is huge. Broadly speaking, a credit derivative is a contract where the payoff depends partly upon the credit-worthiness of one or more commercial or government entities [61]. The total outstanding notional of all credit derivatives in 2004 is estimated to be around 5 trillion dollars [1]. The market is split into two main product types, credit default swaps, which make up around three quarters of the market, and portfolio (or multi-name) credit products, which make up around one fifth of the market. There also exists a growing number of hybrid credit derivative products that are written on more than one asset class, such as EDO's which contain both equity values and credit quality as their underlyings.

Credit derivatives, like all other financial products, can be used to speculate on the market by taking on extra credit risk, or as a hedging tool to hedge away credit risk. Therefore, there is a broad range of end-users of credit derivatives including banks, corporations, hedge funds, insurance and reinsurance houses, and asset managers. Currently the major users of credit derivatives are Europe and North America, with only a small percentage of the market in Asia.

2.1.2 Some Common Credit Derivative Instruments

As stated above, the most commonly used credit derivative is the *credit default swap*, or CDS. One can think of this as a type of insurance contract that is used to transfer the credit risk of the loss of face value of a reference asset until a specified maturity date [1]. There are two parties in a credit default swap, the protection buyer (who is buying protection and selling away credit risk) and the protection seller (who is buying credit risk). Between the initiation of the trade and maturity the protection buyer makes regular payments to the protection seller. If a default event occurs before maturity the protection seller pays the protection buyer the par less the recovery amount of the defaulted reference credit underlying the swap contract.

When one talks about the default of a reference credit, this can actually refer to a host of different credit events. The exact nature of the event that triggers default will be stipulated in the contract of a credit derivative. The most common default events are,

Bankruptcy - the reference firm is unable to pay its debts,

Failure to pay - the reference firm has failed to make any due payments such as bond coupons,

Restructuring - changes to the debt obligations of the reference firm, such as agreeing new terms upon which to pay back existing debt.

From the perspective of building a model of default events, one can think of all of the above default events as essentially the result of a reference firm being in financial distress and unable to service its debt as initially agreed.

As we also stated above, the second most common type of credit derivative are portfolio credit products. It is these types of products that are most relevant to our study as pricing these assumes some kind of default dependence between the reference assets.

One type of portfolio credit product is the *nth to default basket* [3]. This is similar to the CDS as it is possible to take either side of the contract by buying or selling protection. However, now there is more than one underlying reference credit, and pay-out occurs after n of the reference assets have defaulted. Hence, in pricing these instruments one is interested in the dependence between these underlying reference credits - if one credit defaults, will this cause other credits to default before the end of the lifetime of the contract?

Another common type of portfolio credit product is the synthetic CDO or *collateralized debt obligation* [4] [5]. These are also referred to as portfolio default swaps. A synthetic CDO is constructed by taking a portfolio of around 100 names (reference credits) and splitting down the risk of default of the names into three tranches. The first tranche is called the equity tranche and this will normally take around the first 5 percent of losses on the portfolio as the underlying credits begin to default. The next tranche is the mezzanine tranche and this normally takes the next 15 to 20 percent of portfolio losses. Finally, there is the senior tranche which is exposed to all subsequent losses. Investors are paid a spread amount for providing credit protection for a particular tranche based on how much of that tranche is remaining. The spread payments are different depending upon which tranche is being invested in. As the equity tranche is exposed to the first defaults it is therefore the most risky. Investors in the equity tranche receive the largest spread for each dollar of protection they agree to provide.

Again, as with the nth to default basket, correct valuation of CDO's requires making assumptions about the default dependence between the underlying names. An investor in any of the tranches is interested in the probability that given the default of a reference credit, what is the probability of further reference credits defaulting as a result. Some CDO's are actually managed throughout the contract life. This makes is possible for a manager to remove particular names from the basket of credits and replace them other names. Therefore, it is clearly useful for the manager to be able to measure the effect of particular names in the basket defaulting on those that remain, and to remove those that pose most risk to the resilience of the basket to further default events.

2.1.3 Default Correlation

Default correlation is the measurement of the tendency of assets to default together [6]. If assets in a portfolio have a high default correlation then when the assets default, they do so in large groups. Mathematically the default correlation, ρ_D, is expressed as,

$$\rho_D = \frac{p^{AB} - p^A p^B}{\sqrt{p^A(1-p^A)}\sqrt{p^B(1-p^B)}}, \tag{2.1}$$

where p^A is the probability that asset A defaults before some time T, p^B is the probability that asset A defaults before some time T, and p^{AB} is the probability that both asset A and asset B default before time T.

Due to a relative infrequency of default events, empirical analysis of default correlation is difficult and limited [7]. Therefore, to successfully value multi-name credit products, we believe it is important to have an understanding of the underlying factors and determinants that drive default correlation and dependent defaults.

2.1.4 The Structural Approach to Modeling Default Risk

The main idea behind the structural approach to modeling credit risk goes back to the work of Black and Scholes [8], and Merton [9]. In these seminal papers, they showed that one could consider corporate liabilities as contingent claims on the assets of a firm. If one assumes that a firm is financed by equity and a zero coupon bond with face value K and maturity T, then the firm must pay back an amount K to the bond investors at time T. The value of the firm at time T is given by V_T. At maturity, T, there are two possible outcomes for the values of bonds, equity, and hence the firm, depending upon whether the firm has defaulted:

	Assets	Bonds	Equity
No Default	$V_T \geq K$	K	$V_T - K$
Default	$V_T < K$	V_T	0

Therefore, if one were to consider the payoff functions of the bonds and equity at maturity, T, one can think of the bondholders being short a European put option, with strike K, on the assets of the firm, and the equity holders being long a European call option, with strike K, on the assets of the firm.

This way of viewing the capital structure of a firm has been built upon significantly, addressing assumptions above such as the firm being financed by one zero coupon issuance [10], that the firms value is only considered at maturity, and that equity is a *perpetual* call option on the firm's assets.

Currently, one of the most successful structural approaches is that of Moody's KMV [11]. This is based on the basic philosophy of the Merton approach to produce a "distance to default" for each firm. This is a measure of by how many standard deviations the asset value of the firm must move to result in a default, and hence must take account of the asset volatility, firm default point (value of its existing liabilities), and the market value of the firms assets. As it is difficult to measure all these quantities in practice, estimates are made from the equity value and equity volatility, and the idea that the equity is a call option on the firm's assets. Finally a database of historical default events is used to translate the distance to default to a default probability.

The structural approach can then be used to model dependent defaults by assuming some correlation or dependence between the asset value processes of each firm [12]. However, in its simpler forms this approach does not account for direct linkages between firms which may not be obvious from a measurement of the correlation of the firms' asset or equity values. Nor does it account for the numerous reasons for a firm defaulting by surprise, even given apparently healthy balance sheets, such as natural disasters or corrupt management. Of course, these effects can be built in, but not without sacrificing model simplicity and the validity of any model calibration.

2.1.5 The Intensity Based Reduced Form Approach to Modeling Default Risk

While the structural approach is built from an understanding of the capital structure of a firm, the intensity-based reduced form approach assumes nothing about a firm's asset value and corporate liabilities.[1] The idea behind the intensity based approach, initiated by Jarrow and Turnbull in 1995 [13], is to model the statistical properties of the default process. Default is modelled as an exogenous event and the central object in the model is a counting process, N. This is a stochastic process which takes only integer values and counts the number of default events that occur up to some time t. The simplest of these counting processes is the Poisson process (which is described in appendix A), which is governed by an intensity or hazard λ. In reference to a default process, λ is called the default intensity or hazard rate.

Assuming that the hazard rate, λ, has a constant value, let us consider the value, B^T of a zero coupon bond with maturity, T, and zero recovery value given default, i.e., the bond pays one dollar if no default occurs until T and nothing if default occurs before T. Under the Poisson process assumptions, the survival probability is given by,

$$P[\tau > T] = e^{-\lambda T}, \tag{2.2}$$

where τ is the default time of the firm. The value of a corresponding risk free (non-defaultable) bond, $B_0^T = e^{-rt}$, if we assume that interest rates $r > 0$ are constant. Therefore, the value of the risky bond is given by,

$$\begin{aligned} B^T &= B_0^T P[\tau > T] && (2.3) \\ &= e^{-(r+\lambda)T}. && (2.4) \end{aligned}$$

[1] We shall refer to the intensity based reduced form approach as both the reduced form approach and the intensity based approach.

Hence, if there is no recovery value, the yield spread of the risky bond over a risk free bond is exactly equal to the default intensity λ.

As we are interested in an arbitrage-free model of default, this survival probability given above is a risk-neutral measure, i.e, it is not necessarily the same as the real world historical default probability, but comes from the idea of hedging away default risk in the market and is therefore directly related to real market values of credit spread. The corresponding default intensity is called the risk-neutral default intensity or risk-neutral hazard rate.

In reality, the default intensity process cannot be assumed constant and is modelled as some function of time. The survival probability is then given by,

$$P[\tau > T] = e^{-\int_0^t \lambda(s)ds}. \tag{2.5}$$

For example, the hazard rate, $\lambda(t)$ may be stochastic. This is called a Cox process or doubly stochastic process, where N is a Poisson process conditional on the realisation of λ. This idea was introduced by Lando in 1998 [14].

The intensity-based reduced-form approach can be used to model default correlation between firms by including some dependence between the processes underlying the default intensities. For example, there is the direct firm linkage of Jarrow and Yu [27] which models the idea of default contagion and is directly analogous to our intensity based network model of firm defaults.[2] Also, there is the work of Duffie and Singleton [15], in which the default intensity of each individual firm includes some factor which admits common jumps in the intensities of many firms, thereby modelling joint credit events.

2.1.6 Current Models of Default Contagion

Default contagion is the idea that the default of one firm can trigger the default of another, through some dependence between both firms. More broadly, credit contagion is where the deterioration in credit quality of one firm may lead to a deterioration in credit quality of another firm, where neither firm has actually defaulted but both have an increased probability of default. In our study we refer to both of these effects as default contagion as we believe they are essentially manifestations of the same idea - a direct default dependence between firms.

There have been various approaches to modelling default contagion published in recent years. We have already mentioned the intensity based model of looping default

[2] We give more details of the Jarrow And Yu model in section 3.2.

by Jarrow and Yu which we detail in section 3.2. Davis and Lo proposed a simple model of default contagion in 1999 which they refer to as *infectious defaults* [16]. In their model they consider a portfolio of n identical bonds, where each bond may represent some underlying asset in a multi-name credit product such as a CDO. They let $(Z_i, i = 1, ..., n)$ be random variables such that $Z_i = 1$ if bond i defaults and $Z_i = 0$ otherwise. Hence, the number of bonds defaulting is $N = Z_1 + Z_2 + ... + Z_n$. The value of Z_i is determined by the expression,

$$Z_i = X_i + (1 - X_i)\Big(1 - \prod_{j \neq i}(1 - X_j Y_{ji})\Big), \tag{2.6}$$

where, $(i = 1, ..., n)$, $(j = 1, ..., n)$, and X_i and Y_{ij} are independent Bernoulli random variables. Therefore, they have incorporated the idea that bond i may default directly $(X_i = 1)$, or it may be "infected" by the default of bond j : if $X_j = 1$, then the random variable Y_{ji} determines whether infection takes place. This model is then extended by removing the assumption that all n bonds are identical, by making each bond belong to one of m different industry sectors. Bonds in different industry sector default independently, while bonds in the same sector are subject to the default infection mechanism above. Therefore, they are assuming to some extent that industry sector is a measure of default dependence, rather than some direct business link such as a manufacturing supply chain.

Egloff, Leippold, and Vanini propose a model of default contagion where they consider both macro and microstructural interdependencies between reference credits in a portfolio [17]. The macrostructure of dependence relates to a firm's sensitivity to common factors such as changes to industry specific fundamentals. The microstructure relates to the factors that drive default contagion such as direct business linkage between firms or the ownership of firm's debt. In a similar manner to our own approach, the microstructural information is modelled using weighted graphs, where graph nodes represent firms and edges constitute the business interdependence between firms. They model the probability of a firm defaulting in terms of the credit rating of that firm. The rating dynamics are described by a stationary discrete time Markov chain which can be calibrated using rating transition matrices such as those produced by Moody's and Standard and Poor's.[3] Further information on direct firm

[3] Agencies such as Standard and Poor's and Moody's award firms a credit rating which is a measure of the "quality" of a firm and the likelihood that it will be able to service its outstanding debt. A rating transition matrix is a probability transition matrix constructed using historical default data, which measures the probability of a firm's credit rating moving from one level to another level over a given time horizon.

interdependence is obtained from accounting figures and expert knowledge based on firm self assessment.

In their study they propose different possible graph structures of interdependence, such as random graphs and graphs centred on perhaps one or two nodes, and assess the impact of these different structures on the distribution of losses on the portfolio of credits. They find that the microstructural dependence impacts only the very far tails of a loss distribution, increasing the probability of very large portfolio losses.

A similar approach to that of Egloff, Leippold, and Vanini is the model of cyclical correlations, credit contagion, and portfolio losses by Giesecke and Weber [18]. They consider both the cyclical default correlation due to the dependence of firms on common economic factors, and also credit contagion associated with local interaction of firms with their business partners. Firms are represented as nodes on a lattice network structure, where each firm is distinguished by a "liquidity state" which measures the ability of that firm to generate cash and honour its obligations. A firm migrates from one liquidity state to another with an intensity that is proportional to the number of business partners in the opposite state.

Their model predicts that average losses on large portfolios of financial positions are dominated by the cyclical oscillations in the fundamentals of an economy. The effect of the default contagion mechanism is to produce additional fluctuations of losses around their averages. They also find that the probability of large losses depends on how connected the business partner network is - the less connected the business network, the greater the probability of a large loss.

2.2 Complex, Social, and Small World Networks

A network or graph is a set of items, called vertices or nodes, that have connections between them called edges. In a directed graph or network the edges have an associated direction from one vertex to another. The degree of a network is the number of edges connected to a vertex. A directed graph has an in-degree and an out-degree for each vertex, which are the numbers of in-coming and out-coming edges respectively.

Examples of systems that can be modelled by networks include the Internet, the World Wide Web, social networks (which we discuss in more detail later on), citation networks between papers, power grids, air travel routes, neural networks, blood vessels, predator-prey interactions between species in a freshwater lake, molecular regulatory network which controls the mammalian cell cycle, and many, many more. In recent years there has been rapid growth in the study of the function and structure

of networks [47][20]. Because many of the networks mentioned above are so large, much of the recent study has considered these networks to be complex systems, where large-scale statistical analysis and descriptions are more appropriate than exact details of which node is connected to which. In fact, some networks, such as the entire World Wide Web, are impossible to analyse by eye and the only way they can be described is by their statistical properties. That said, drawing a picture of a network still remains one of the most powerful tools in network analysis for all but the very largest of structures

Not only is the topology of a network of interest. A lot of recent research has included the evolution of networks through time, and the development of graph-building algorithms whose simple results capture the important properties of far more complex structures [19].

One particular area of study that has attracted much interest is social networks and the evolution of social networks over time. A social network is a network modeling the acquaintance between people. A seminal work by Watts and Strogatz in 1998 looked at various features of social networks and proposed a simple model of a static social network [21]. Within this paper they also coined the term "small world" networks by making an analogy with the small world phenomenon - also known as six degrees of separation - in which everyone on the planet is connected to everyone else via a maximum of five other people. In chapter 5 we utilise the graph measures developed in the analysis of small world networks for the study of our own network structures and also study the simple network construction algorithm developed by Watts and Strogatz in their analysis of social networks. We also look in more detail in sections 12.7 at small world networks themselves with particular reference to the Watts definition of what exactly constitutes a small world network [46].

2.3 Use of Networks in Financial Modeling

It is easy to conceive that many entities in finance can be represented as vertices, such as stocks, firms, investors, financial employees, etc. Then, the relationships between these entities could denote the edges, and thence, we would have created some form of network. Indeed, the application of networks to the study of financial markets is not new and over the years has taken many different forms.

One example is the creation of a network of corporate boards linked by shared board directors (directors on the board of more than one firm). Battiston, Weisbuch and Bonabeau study decision making in such a network, where board directors are

engaged in dynamics based on "herd behaviour", and influence between the boards takes place through the shared directors [22]. They propose agent-based models which simulate the effect of different decision-making processes between the board members, and they study the effect of different network structures between the firms, and the impact of both of these factors on the probability that boards will make the same decisions.

Another recent work by Battiston, Garlaschelli and Caldarelli considers a network in which vertices are companies and a directed edge between vertices denotes ownership of shares in that company, i.e., a company is linked to another company if it owns stock of that company [23]. Studying data from one Italian and two US share indices, they introduce two quantities, HI (Holder Index), and SI (Stock Index). These measure the effective number of stocks controlled by a holder, and the number of effective holders of a stock, respectively. Therefore, one can consider these values analogous to the in-degree and out-degree of a vertex, as they measure the effective number of in-going and out-going links. By studying the distributions of HI and SI they characterize the ownership concentrations of stocks, and the power of stock holders, and also describe the structure of the markets themselves. This work may have important applications to the study of default dependence between firms as is reasonable to view the distribution of stock ownership between firms as some measure of the stability of a network to default events, and the likelihood of default contagion.

Before we move onto the next idea we need to define some more terms from graph theory. A *path* is a sequence of vertices, such that from each of the vertices in the sequence there is an edge to the successor vertex. A *tree* is a graph in which there is only one path connecting each pair of vertices. A graph is *connected* if all vertices can be reached by all other vertices via a path. Given a graph A, a subgraph, B, of graph A is a graph such that every edge of graph B is also an edge of graph A. An edge *weight* is a value associated with an edge, such as the strength of a bond, or the distance between two towns. Now, given a undirected, connected graph G, a *spanning tree* of G is a subgraph of G, that is a tree, and that contains all the vertices of G. Finally, a *minimum spanning tree* is a spanning tree, but it has weights or lengths associated with the edges, and the total weight of the tree (the sum of the weights of its edges) is at a minimum. There are many algorithms to find the minimum spanning tree of a graph, and their application is wide, including the standard example of finding the cheapest way to connect many cities by phone line.

Micciche, Bonanno, Lillo, and Mantegna create minimum spanning trees using the correlation of both asset return series *and* volatility time series between a portfolio of

US equities, where edge weight is given by the correlation value [24]. These minimum spanning trees can then be used to graphically assess the relationships between firms in terms of their correlations. However, whether they can be used as a measure of default dependence between firms depends on how valid one considers the assumption that default dependence can be measured by the correlation between asset returns or volatility.

Onnela et al take an alternative approach where in their methodology, again based on the correlations between asset returns, they construct a more general graph structure which they call a dynamic asset graph [25]. Distance between two stocks is defined as $d_{ij}(t) = \sqrt{2(1 - \rho_{ij}(t))}$, where, here, $\rho_{ij}(t)$ is the correlation between the asset returns of stocks i and j over a rolling window of time up to t. The graph is then constructed by adding edges in ascending order of distance up until some cut-off level. Therefore, there is no reason for their method to necessarily result in the construction of a tree, or even one connected structure. This appears to be a fairer representation of the correlation (and perhaps dependence) between stocks, over that of the minimum spanning tree, as the more general graph structure is not forcing the "acceptance" of edges, may have been the case to satisfy the minimum spanning tree criterion. However, the resulting structure may be less intuitive graphically than the minimum spanning tree and may be more suitable to statistical analysis.

Kullmann, Kertesz, and Kaski construct *directed* network structures from asset return correlations by measuring the correlations as a function of the time shift between pairs of stock return time series [26]. For this they use tick-by-tick data in a hope to identify any leading or lagging between price returns and hence any dependence between firms. They do find a weak but discernible effect where maximum correlations appear at nonzero time shifts. Again, whether a dependence between stock price changes is a good measure of default dependence is unknown, but this approach could certainly act as a guide to the construction of a default dependence network between firms.

Finally, there are the network based default models of Egloff, Leippold, and Vanini, and Giesecke and Weber which we referred to in section 2.1.6. Like are our own models, they represent the default dependence between firms as a network and study the effect of changing the network structure on the default properties of a portfolio of credit names.

2.4 Why an Intensity Based Network Model of Default?

Even though, as we have shown, there has been a lot of research into modeling the default dependence between firms, there does not exist any particular method that has been universally adopted by those working in the field in the same way, for example, as the Black-Scholes model has been adopted by those pricing simple equity derivatives. We believe that much of the reason for this is the current lack of understanding what factors actually drive the default behaviour of a portfolio of credits.

The reduced form intensity based model of firm default is appealing in its relative simplicity, as it does not require the complex modeling of capital structure or rating agency predictions on a firm's "health", etc. This simplicity allows it to be easily applied as the basis for a model in which the default of many firms is of interest. It is also appealing as it does not try and determine exactly *why* a firm has defaulted, but rather models the statistical behaviour of *when* a default might occur.

Using a network to represent the default dependence between firms allow us to, firstly, view graphically, the structure of the default dependence. This is appealing when one wishes to make assessments of the impact of changes to a portfolio dependence structure, or when comparing portfolios of credit. Secondly, we are able to take advantage of the current research into complex networks and the analysis of networks using statistical techniques.

By combining basic ideas from graph theory, current research into the analysis of complex networks, and a simple intensity based default mechanism, we believe that we have the basis of a powerful yet simple approach to understanding the response of a basket of reference credits to default events and credit contagion effects.

Chapter 3

Model Overview

In this chapter we describe the reduced form default model at the centre of our study and introduce the idea of the default matrix. We also show the parallels between our model and that of a model of looping default by Jarrow and Yu.

3.1 Model Description

We assume that a firm can be modelled as a node on a directed network, where the links between each of the nodes represent a dependence between the firms, see figure 3.1.

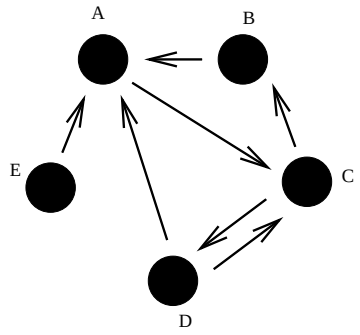


Figure 3.1: Example of a simple directed network of 5 nodes.

This network of firms can be represented by an adjacency matrix $\underline{\underline{A}}$. For the network given in Figure 3.1 the adjacency matrix is:

$$\underline{\underline{\mathbf{A}}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where $A_{ij} = 1$ if a link runs from firm j to firm i, and $A_{ij} = 0$ where there is no link.

Associated with each firm, i, is a default intensity λ_i. This is the Poisson intensity for firm i to default before any other defaults have taken place. Each link is weighted by an amount ρ_{ij}. This is the amount by which the default intensity of node j will be increased, given the default of node i, i.e., it is a measure of the dependence of node j upon the default of firm i, see Figure 3.2.

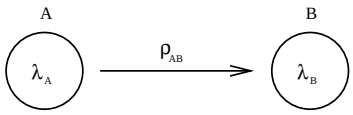


Figure 3.2: Illustration of the default dependence of node B on node A.

If node A in Figure 3.2 were to default, then:

$$\lambda_B \mapsto \lambda_B + \rho_{AB}.$$

We can therefore create what we term a default matrix, $\underline{\underline{D}}$. This represents all of the default intensities within a network of companies. The elements d_{ii} give the initial default intensities of each firm, while elements $d_{ij,i\neq j}$ give the default dependence. For the network given in Figure 3.1 the default matrix is

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda_A & 0 & \rho_{AC} & 0 & 0 \\ \rho_{BA} & \lambda_B & 0 & 0 & 0 \\ 0 & \rho_{CB} & \lambda_C & \rho_{CD} & 0 \\ \rho_{DA} & 0 & \rho_{DC} & \lambda_D & 0 \\ \rho_{EA} & 0 & 0 & 0 & \lambda_E \end{pmatrix}.$$

An important point to note is that in our model, defaults of firms are independent of each other in that the Poisson processes relating to each firm are independent of each other. Dependency is then introduced via the default dependency values, p_{ij}. Hence, the dependency is included only in a "consecutive" sense - given the default of one firm, remaining firms may then have adjusted default intensities - rather than the "simultaneous" dependency that would be implied by correlated Poisson processes.

3.2 Jarrow and Yu model of Looping Default

The basic mechanism behind our model is the same as a model of looping default of counter-party risk under the reduced-form framework by Jarrow and Yu [27]. Given two firms A and B, they postulate that if the default probability of firm A is strongly

effected by the default of firm B, the the calculation of the bond price of A entails knowing the distribution of the default time of firm B. However, if firm B holds a significant amount of debt issued by A, then the distribution of the default time for B would depend on that of A. Therefore, as given above, the default intensities of each form are described by:

$$\lambda_A(t) = a_1 + a_2 1_{t \geq \tau^B} \tag{3.1}$$

$$\lambda_B(t) = b_1 + b_2 1_{t \geq \tau^A}, \tag{3.2}$$

where τ^i is the default time of firm i. In their study they stated that because the distributions of default times were defined recursively through each other, they could only be obtained explicitly in special cases. Therefore, by stating that looping defaults were unlikely to occur in practice, they moved on to a simplified case called the primary-secondary framework. Here a restriction is imposed that there are now two types of firms; primary firms - whose default intensities do not depend directly on the default of any other firms, and secondary firms - whose default intensities do depend on the status of primary firms. Hence, If firm A is primary, and firm B is now secondary, their intensities are given by:

$$\lambda_A(t) = a_1 \tag{3.3}$$

$$\lambda_B(t) = b_1 + b_2 1_{t \geq \tau^A}. \tag{3.4}$$

Jarrow and Yu then apply the primary-secondary framework to the pricing of various instruments including bonds and credit default swaps.[1]

In our model we retain the very general approach of many firms which can all have an unequal default dependence upon one another.

[1] A very recent working paper by Yu, [28], has now addressed the inability in the Jarrow and Yu paper to provide an explicit solution for the distribution of default times in the general case, by using a technique called total hazard construction.

Chapter 4

Developing a Mathematical Framework for Measuring Expected Time of the kth to Default

In this chapter we develop a mathematical framework that allows us to calculate, for a network of n_0 nodes, the expected time of the kth default, for $1 \leq k \leq n_0$. We work through from the simplest case where all nodes are unconnected, to a case in which the graph is fully connected, and on to the general case for any undirected network.

4.1 Independent Defaults - The Death Process

The simplest possible case of a network of firms is that the firms are identical and all the links between them have a weight of zero, i.e., the graph is completely unconnected. The distribution of default time for each firm will be independent of the distribution of the default time for the other firms, and all firms have the same default intensity, λ.

Therefore, the unconnected network of firm defaults can be modelled as a particular generalised Poisson process called a death process. In this process, a firm present at time t has probability $\lambda\Delta t + o(\Delta t)$ of defaulting in the small interval $[t, t + \Delta t]$. This probability is the same for all of the firms and is independent of the age of the firm. It is the opposite of the linear birth process in which an individual has a probability $\lambda\Delta t + o(\Delta t)$ of splitting into two (or giving birth) in the interval $[t, t + \Delta t]$.[1] Figure 4.2 gives an illustration of a realisation of the death process.

[1]For an example of the birth process and other generalised Poisson processes see [29].

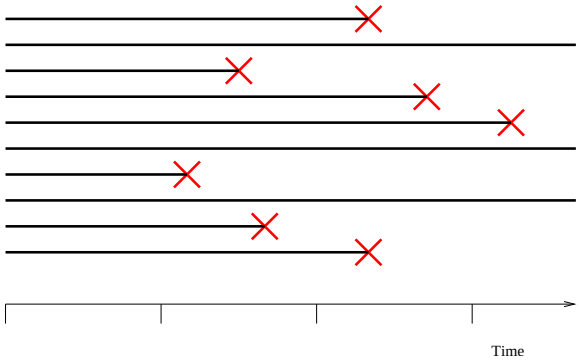


Figure 4.1: Example of a realisation of the death process.

Initially, we can make a deterministic analysis of the problem for an unconnected network of firms, where n_0 is the number of firms at $t = 0$ and $n(t)$ is the number of firms at time t and is very large. As $n(t)$ is large we can treat it as a continuous function of t. Therefore, $\lambda n(t)\Delta t$ defaults occur in the interval $[t, t + \Delta t]$, and hence,

$$n'(t) = -\lambda n(t), \tag{4.1}$$

$$n(t) = n_0 e^{-\lambda t}, \tag{4.2}$$

where n_0 is the number of individuals at time $t = 0$.

Analysing the system as a stochastic process, let the random variable $N(t)$ be the number of firms at time t. Also, let prob$\{N(t) = i\}$ be denoted as $p_i(t)$. Using the fact that the superposition of i Poisson processes, each with rate λ, is just a new Poisson process with rate $i\lambda$ (see Appendix A, section A.3), the probability of a default in the interval $[t, t+\Delta t]$ is therefore $i\lambda\Delta t+o(\Delta t)$. From here we can construct the forward equations

$$p_i(t + \Delta t) = \text{prob}\{N(t) = i \text{ and no default occurs in } [t, t + \Delta t]\} \tag{4.3}$$

$$+\text{prob}\{N(t) = i + 1 \text{ and one default occurs in } [t, t + \Delta t]\} \tag{4.4}$$

$$+o(\Delta t), \tag{4.5}$$

$$= p_i(t)(1 - i\lambda\Delta t) + p_{i+1}(t)(i + 1)\lambda\Delta t + o(\Delta t). \tag{4.6}$$

This then leads to

$$p_i'(t) = -i\lambda p_i(t) + (i + 1)\lambda p_{i+1}(t) \tag{4.7}$$

for $i = n_0, n_0 - 1$..... The initial condition for (4.7) is

$$p_i(0) = \delta_{in_0} = \begin{cases} 1 & \text{if } i = n_0 \\ 0 & \text{if } i \neq n_0, \end{cases}$$

where δ_{ij} is the Kronecker delta symbol.

To solve (4.7), we introduce the probability generating function

$$G(z,t) = \sum_{i=0}^{\infty} p_i(t) z^i \tag{4.8}$$

which satisfies

$$\frac{\partial G(z,t)}{\partial t} + \lambda(z-1)\frac{\partial G(z,t)}{\partial z} = 0, \tag{4.9}$$

where

$$G(z,0) = z^{n_0}. \tag{4.10}$$

We can now solve (4.9) using the method of characteristics to give

$$\begin{aligned} G(z,t) &= ((z-1)e^{-\lambda t}+1)^{n_0}, & (4.11)\\ &= \sum_{i=0}^{n_0} \binom{n_0}{i} (1-e^{-\lambda t})^{n_0-i} (e^{-\lambda t})^i z^i. & (4.12) \end{aligned}$$

Hence, the probability of the number of unconnected firms at time t is

$$\begin{aligned} p_i(t) &= \binom{n_0}{i} (1-e^{-\lambda t})^{n_0-i} (e^{-\lambda t})^i \qquad \text{for } 0 \le i \le n_0 & (4.13)\\ &= 0 \qquad \text{otherwise.} & (4.14) \end{aligned}$$

Below we give the probabilities for $i = 0, ..., 3$ firms at time t for a small completely unconnected network of 3 firms, i.e., $n_0 = 3$,

$$\begin{aligned} \text{prob}\{N(t)=0\} &= (1-e^{-\lambda t})^3 & (4.15)\\ \text{prob}\{N(t)=1\} &= 3(1-e^{-\lambda t})^2 (e^{-\lambda t}) & (4.16)\\ \text{prob}\{N(t)=2\} &= 3(1-e^{-\lambda t})(e^{-\lambda t})^2 & (4.17)\\ \text{prob}\{N(t)=3\} &= e^{-3\lambda t}. & (4.18) \end{aligned}$$

Now, we are in a position to write down the first passage times, i.e., the expected times for a certain number of firm defaults to occur. This is straightforward as $N(t)$ cannot increase. Let T_a be the time at which state a is first occupied ($a < n_0$), where here we refer to a state in terms of the number of firms still active, see Figure 4.2. Therefore,

$$\text{prob}\{N(t) \le a\} = \text{prob}(T_a \le t). \tag{4.19}$$

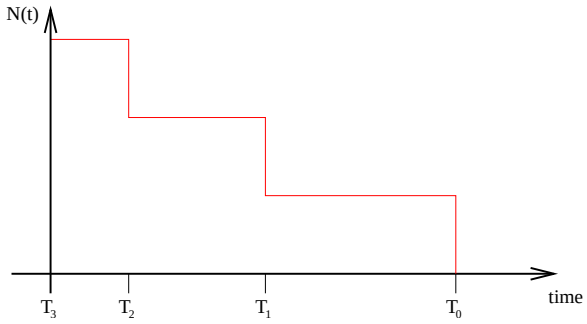


Figure 4.2: Illustration of the variable T_a for $n_0 = 3$.

Hence, for our example of an unconnected network with $n_0 = 3$, the probability that the number of firms in existence at time t will be less than or equal to 1, is given by

$$\begin{aligned} \text{prob}(T_1 \leq t) &= \text{prob}\{N(t) \leq 1\} && (4.20) \\ &= \text{prob}\{N(t) = 1\} + \text{prob}\{N(t) = 0\}. && (4.21) \end{aligned}$$

Substituting in the values we calculated above we have

$$\text{prob}(T_1 \leq t) = 1 - 3e^{-2\lambda t} + 2e^{-3\lambda t} = \mathcal{F}_1(t), \quad (4.22)$$

where $\mathcal{F}_1(t)$ is the cumulative probability distribution function for $N(t) \leq 1$. Therefore, the expected time to be in state $a = 1$ is given by

$$E(T_1) = \int_0^\infty t f_1(t) dt, \quad (4.23)$$

where $f_1(t)$ is the probability density function for $N(t) \leq 1$. Hence,

$$\begin{aligned} E(T_1) &= \int_0^\infty t[6\lambda e^{-2\lambda t} - 6\lambda e^{-3\lambda t}]dt, && (4.24) \\ &= \frac{5}{6\lambda}. && (4.25) \end{aligned}$$

In general, we can write

$$\begin{aligned} \text{prob}(T_a \leq t) &= \text{prob}\{N(t) \leq a\}, && (4.26) \\ &= \sum_{i=0}^{a} p_i(t) && (4.27) \end{aligned}$$

and thence,

$$E(T_a) = \int_0^\infty t \left(\sum_{i=0}^{a} \frac{dp_i(t)}{dt} \right) dt. \quad (4.28)$$

By expanding the summation in (4.28), we have

$$\frac{dp_a(t)}{dt} = -a\lambda p_a(t) + (a+1)\lambda p_{a+1}(t) \qquad \text{for } a \geq 0. \tag{4.29}$$

Summing these, we have

$$\sum_{i=0}^{a} \frac{dp_i(t)}{dt} = (a+1)\lambda p_{a+1}(t). \tag{4.30}$$

into equation 4.28. Hence, the expected time to be in state a can be written as

$$E(T_a) = (a+1)\lambda \int_0^\infty t p_{a+1}(t)dt, \tag{4.31}$$

where

$$p_{a+1}(t) = \binom{n}{i}(1-e^{-\lambda t})^{(n_0-(a+1))}(e^{-\lambda t})^{(a+1)}. \tag{4.32}$$

We now have an expression to calculate the expected time for k defaults events to occur for a completely unconnected network of firms. Of course, looking at a completely unconnected network is far too great a simplification of our model, but analysis of the problem does provide useful insight into the best method to tackle the general case of any undirected network structure.

4.2 Independent Defaults - The Summation Method

By utilising the properties of Poisson processes and generalisations of Poisson processes, such as the superposition of independent Poisson processes, it is possible to attack the problem of finding the expected time for k defaults to occur much more simply. This simpler method, which we introduce here, will become the central building block for a general approach for any network type.

In Section 4.1 we introduced T_a as the time at which state a is first occupied, where a refers to the number of firms remaining in the network. It is possible to think of T_a as the sum of independent random variables. As the waiting time of a Poisson process is exponentially distributed, we let z_i be the length of time spent in state i, where z_i is an exponentially distributed random variable with parameter $i\lambda$. Again, λ is the default intensity of each firm in the network. Referring to the properties of a Poisson process in Appendix A, each of the different z_i for each of the different states must be independent of each other. Hence, we can deduce that

$$T_a = z_{n_0} + z_{n_0-1} + ... + z_{a+1}, \tag{4.33}$$

where n_0 is the initial number of firms in the network. Therefore, the expected time to be in state a is just the sum of the expected values of each of the z's,

$$E(T_a) = \frac{1}{\lambda} \sum_{j=a+1}^{n_0} \frac{1}{j}, \tag{4.34}$$

and similarly, the variance is

$$V(T_a) = \frac{1}{\lambda^2} \sum_{j=a+1}^{n_0} \frac{1}{j^2}. \tag{4.35}$$

Clearly, considering the expected time to be in a particular state as the sum of independent random variables allows for a very intuitive description of the default process. We now move on to look at a more complicated network - the completely connected network.

4.3 A Completely Connected Network of Firms

A completely connected network, or more correctly a completely connected directed graph, is one in which all possible links that can be made are made. Hence, all nodes are linked to all other nodes. Building on from our previous simplistic model in which all firms are independent of each other, we now consider a setup in which all firms have the same default intensity λ but now are situated on a completely connected directed graph with an equal edge weighting of ρ. Therefore, if any firm in the network were to default, *all* of the remaining firms would be affected, and their default intensity would be increased by an amount ρ; see Figure 4.3. Essentially, what is occurring here is that upon default of a firm, the intensity of the combined Poisson process increases by ρ and decreases by λ.

As all firms have an identical dependence upon one another we can write down the expected time to be in state a,

$$\begin{aligned} E(T_a) &= \frac{1}{n_0\lambda} + \frac{1}{(n_0-1)(\lambda+\rho)} + & (4.36) \\ &= \sum_{i=0}^{n_0-(a+1)} \frac{1}{(n_0-i)(\lambda+i\rho)}, & (4.37) \end{aligned}$$

where a, which lies in the range $0 \leq a < n_o$, refers to the number of remaining firms, and n_0 is the number of firms initially. Similarly,

$$V(T_a) = \frac{1}{(n_0\lambda)^2} + \frac{1}{((n_0-1)(\lambda+\rho))^2} + ... \tag{4.38}$$

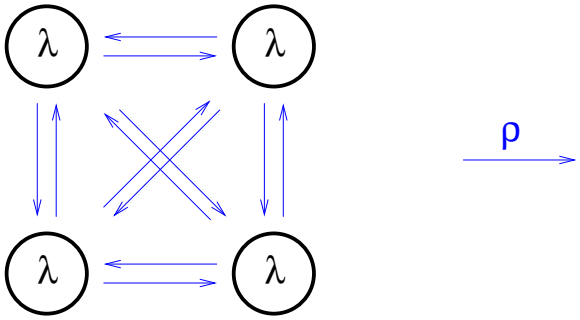


Figure 4.3: Example of a completely connected undirected graph of 4 firms with identical default intensities λ and default dependencies ρ.

$$= \sum_{i=0}^{n_0-(a+1)} \frac{1}{(n_0-i)((\lambda+i\rho))^2}. \tag{4.39}$$

Now, we have looked at the two extremes - a completely unconnected network of firms and a completely connected network of firms. Based upon this method of summing the independent exponentially distributed random variables of the time to be in each state, we can move onto the general case of any network.

4.4 Summation Method for any Network Structure

Here we show that using a development of the summation method detailed above, we are able to find the expected time to be in state a, where a is the number of firms remaining, for any structure of network. Our method also allows all firm default intensities, λ_i, and default dependencies, ρ_i to be firm specific, rather that the same for each firm as is required in the two simple cases illustrated above.

To recap, all information about the default process is held in a default matrix. Consider a simple general network of 3 nodes, A, B, and C, illustrated in Figure 6.4. The default matrix for this graph (Figure 6.4) is

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda_A & \rho_{AB} & \rho_{AC} \\ \rho_{BA} & \lambda_B & \rho_{BC} \\ \rho_{CA} & \rho_{CB} & \lambda_C \end{pmatrix}.$$

Now that we can no longer treat the firms as identical (as was the case for the completely unconnected and completely connected networks considered earlier) we need to consider the individual probability that each of the firms will default. For the

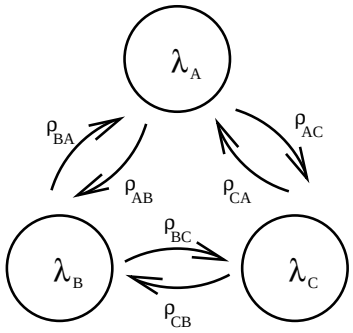


Figure 4.4: Example of a completely connected undirected graph of 3 firms, A, B, and C, with individual default intensities and default dependencies.

example shown above, initially, we have $n_0 = 3$ firms and no defaults have taken place. We can treat this process as the superposition of 3 independent Poisson processes with default intensities, λ_A, λ_B and λ_C. This is itself a Poisson process with default intensity λ where

$$\lambda = \lambda_A + \lambda_B + \lambda_C; \tag{4.40}$$

see Appendix A for a proof.

The time Z to the first default of this new combined Poisson process is again an exponentially distributed random variable with parameter λ, and p.d.f. given by $\lambda e^{-\lambda t}$. We can denote by Z_A, Z_B, and Z_C the default times for each of the separate Poisson processes for firms A, B, and C, with p.d.f's $\lambda_A e^{-\lambda_A t}$ etc. Therefore, the expected time of the first default for all 3 processes combined is $Z = \min(Z_A, Z_B, Z_C)$.

Crucial to solving the general network case, we can now find the probability that each of the firms A, B and C was the first of the 3 to default. Given that the first default occurs at time t, i.e., $Z = t$, the probability that firm A was the first of the 3 firms to default, is the probability that it defaults at time t and firms B and C default at some time greater than t,

$$\begin{aligned} \text{prob}\{Z_A &= t, Z_B > t, Z_C > t | Z = t\} && (4.41)\\ &= \frac{\lambda_A e^{-\lambda_A t}.e^{-\lambda_B t}.e^{-\lambda_C t}}{\lambda e^{-\lambda t}} && (4.42)\\ &= \frac{\lambda_A}{\lambda}. && (4.43) \end{aligned}$$

Hence, the probability of firm A defaulting first out of the 3 firms is just the ratio of the default intensity of firm A over the sum of the intensities of firms A, B and C.

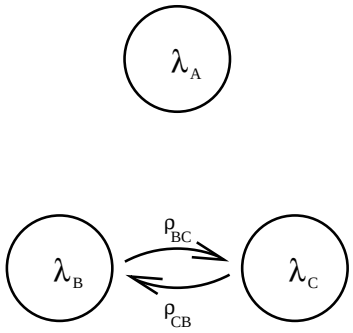


Figure 4.5: Plot of network structure for connecting firms A, B, and C.

Once firm A has defaulted, the default intensities of firms B and C are increased by the values of ρ_{AB} and ρ_{AC}, respectively. Because the probability for B and C to default is independent of the default time of firm A, we can now construct a new combined Poisson process with default intensity equal to the sum of the new adjusted intensities of firms B and C. From here, we can calculate again the new probabilities that either firm B or C will be the first to default.

Therefore, we can build a up tree of all the possible combinations of the order in which the firms can default, along with the associated probability of each state being reached. To see this in practice we consider the following simple example. Three firms A, B and C, sit on the network described by the following default matrix:

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda_A & 0 & 0 \\ 0 & \lambda_B & \rho_{BC} \\ 0 & \rho_{CB} & \lambda_C \end{pmatrix}.$$

This simple network structure is sketched in Figure 4.5. As a further simplification we set $\lambda_A = \lambda_B = \lambda_C$ and $\rho_{BC} = \rho_{CB}$. This will simplify the mathematics somewhat, but not detract from illustrating the method.

Figure 4.6 shows the default tree for this simple 3 - firm example. Each edge in the default tree is the probability of moving from one state to another. State 0 indicates that no defaults have occurred, while state BA is the state in which B has defaulted and then A defaults etc. Given the result in (4.43), the probability of moving from state 0 to states A, B, or C is simply the ratio of the individual default intensities associated with each independent Poisson process to the total default intensity for the combined process, i.e., $\frac{\lambda}{3\lambda} = \frac{1}{3}$.

Once, state A, B, or C has been reached, we then calculated the new probabilities for changing state from the two remaining independent Poisson processes from the

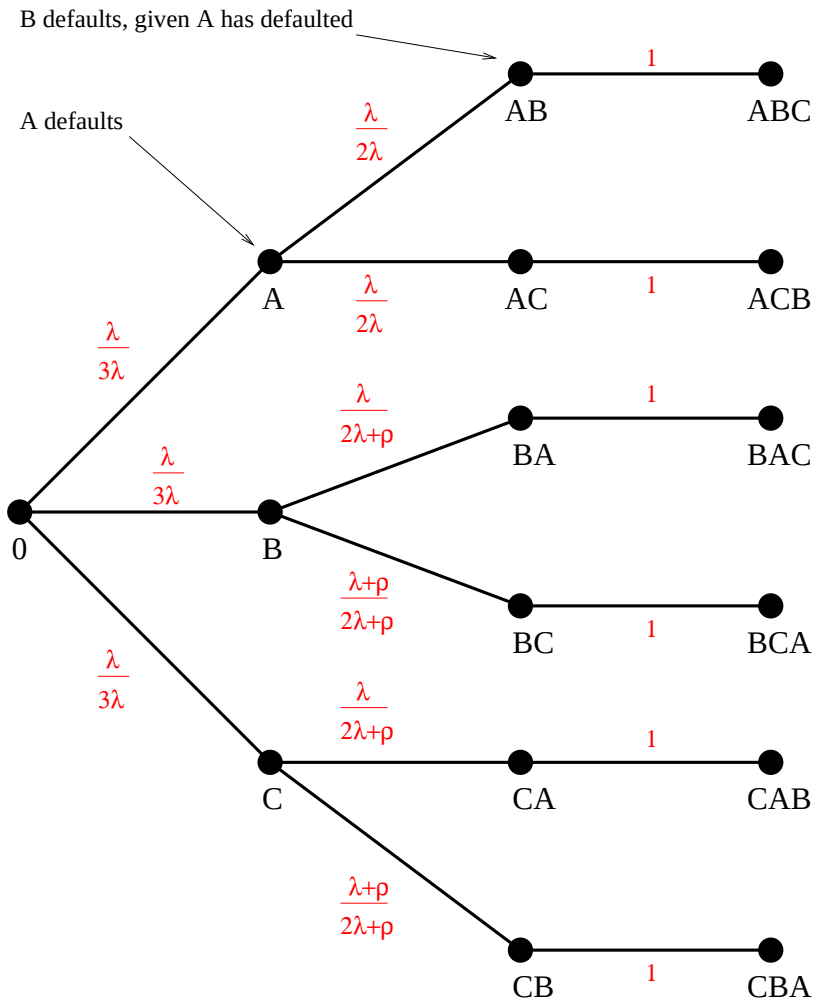


Figure 4.6: Plot of default tree structure for the network shown in Figure 4.5

adjusted default intensities for the two remaining Poisson processes relating to each of the two surviving firms.

It is then straightforward to calculate the expected time for a certain number of defaults to occur. As before, T_a is the time at which state a is first occupied, where a is the number of firms remaining, $(a < n_o)$. Then, for the example above,

$$E(T_2) = \frac{1}{3\lambda}, \tag{4.44}$$

as no companies have defaulted, just consider the sum of the initial individual intensities. The expected time to reach the state where there is only one firm remaining is given by

$$\begin{aligned} E(T_1) &= E(T_2) + \frac{1}{3}.\frac{1}{2\lambda} + \frac{1}{3}.\frac{1}{(2\lambda+\rho)} + \frac{1}{3}.\frac{1}{(2\lambda+\rho)} && (4.45)\\ &= \frac{1}{2\lambda} + \frac{2}{3(2\lambda+\rho)}. && (4.46) \end{aligned}$$

Note that this is just the expected time to reach the state where there are two firms remaining, plus the intensities of the three possible resulting Poisson processes multiplied by the probabilities that each of these processes will occur. Similarly, the expected time to reach the state where there are no firms in existence is just,

$$\begin{aligned} E(T_0) &= E(T_1) + \frac{1}{3}\Big(\frac{\lambda}{2\lambda}.\frac{1}{(\lambda+\rho)} + \frac{\lambda}{2\lambda}.\frac{1}{(\lambda+\rho)}\Big)\\ &+ \frac{1}{3}\Big(\frac{\lambda}{(2\lambda+\rho)}.\frac{1}{(\lambda+\rho)} + \frac{(\lambda+\rho)}{(2\lambda+\rho)}.\frac{1}{\lambda}\Big)\\ &+ \frac{1}{3}\Big(\frac{\lambda}{(2\lambda+\rho)}.\frac{1}{(\lambda+\rho)} + \frac{(\lambda+\rho)}{(2\lambda+\rho)}.\frac{1}{\lambda}\Big). \end{aligned} \tag{4.47}$$

4.4.1 Summary of Method Used to Construct Default Tree

Using the above method of summing the intensities for remaining independent Poisson processes and generating a tree of probabilities for all the possible combinations of defaults to occur, it is possible to find the expected time for *any* number of defaults to occur for *any* directed network structure between the firms. Figure 4.7 shows a node Y being fed by node X and itself feeding nodes $Z_1, Z_2, \ldots Z_N$. Each node represents a state of the system where a certain number of defaults have occurred in a particular order. In general, the expected time to be in state Y is just

$$\begin{aligned} &\frac{1}{\text{sum of the intensities of all processes associated with leaving state } Y}\\ = \;& \frac{1}{\sum_{i=1}^{N} \lambda_{Y\to Z_i}}, \end{aligned} \tag{4.48}$$

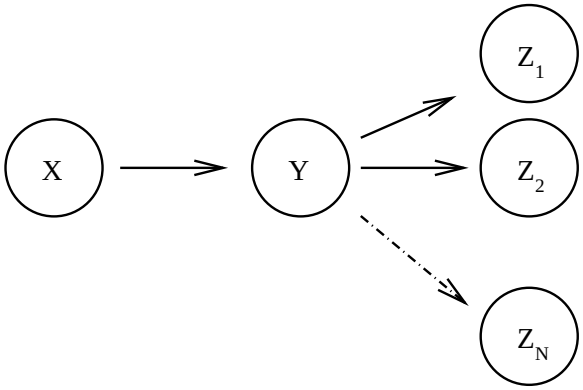


Figure 4.7: Illustration of general case default tree with one vertex being fed by one other, and itself feeding N others.

where $\lambda_{a\to b}$ is the intensity associated with the branch on the default tree that represents a movement from state a to state b. The probability of moving from state Y to state Z_i, *given* a default event is

$$\begin{aligned} & \frac{\text{intensity associated with moving from state } Y \text{ to state } Z_i}{\text{sum of the intensities of all processes associated with leaving state } Y} \\ = \; & \frac{\lambda_{Z_i}}{\sum_{i=1}^{N} \lambda_{Y\to Z_i}}. \end{aligned} \tag{4.49}$$

Then, from a given initial state, to calculate the expected time for a certain number of default events to occur, one just moves along the tree summing the expected times to be in all states on the default tree that lead up to, but not including, states in which that number of defaults has already occurred, multiplied by the associated probabilities of reaching each of those states given a default event.

4.5 A Binary Based General Algorithm Approach

It is useful to be able to provide a general algorithm that one must apply to calculate the expected time for a certain number of defaults to occur. This is sensible both from the mathematical understanding point of view and also from a computational point of view, when dealing with larger numbers of firms and more complex network structures.

Before we continue we need to introduce some notation for the default state of a set of firms. The default state of a portfolio of n_0 firms can be represented as a bit string of length n_0. Counting from the left of the bit string, each bit refers to whether firm, A, B, C, D, etc., in that order, has defaulted. A zero bit indicates that the

firm has not defaulted, a one bit indicates that the firm has defaulted. For example, for a portfolio of 5 firms labelled A, B, C, D and E, the bit string 00101 indicates that firms A, B, and D have not defaulted, and firms C and E have defaulted. For a portfolio of 8 firms, the bit string 00101101 represents the state in which firms C, E, F, and H have defaulted while the remaining firms, A, B, D, and G, are yet to default.

Now to demonstrate our general algorithm we again consider just the simple case of a network of three firms A, B, and C. The particular default dependence structure we will use in our demonstration is given in Figure 4.5.

Because of the nature of the default mechanism, the expected time to be in each binary state is independent of the order of defaults that occured to reach that state, e.g., the expected time to be in a state 011 in which firms B and C have defaulted is the same whether it came from either state 001, where firm C was the first to default, or state 010, where firm B was the first to default, see Figure 4.8. Note this

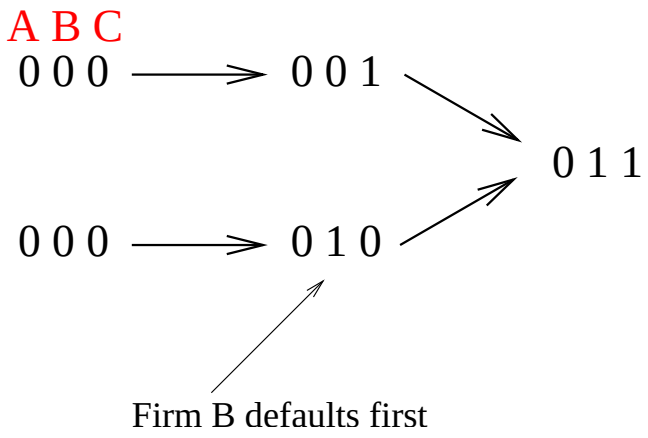


Figure 4.8: Two possible paths leading to state 011.

is possible because the default mechanism is such that the default intensities of firms are incremented by some constant value given the default of a neighbouring firm, i,e., if a firm were connected to two others, it does not matter in what order the two neighbours default, the resulting change to the default intensity of the reference firm *after both firms had defaulted* is the same. Therefore, for our model, each state (001, 101, etc.) is a Markov state, as the expected time to leave that state is independent of the path to reach it. Hence, the default tree we have used up to now to represent the possible states of a portfolio of firms given different combinations of default events (i.e., BA and ABC etc.), such as the one given in Figure 4.6, can be simplified using

our binary notation such that we do not distinguish between states AB and BA etc.[2] We give this new binary default graph for a portfolio of 3 firms in Figure 4.9.

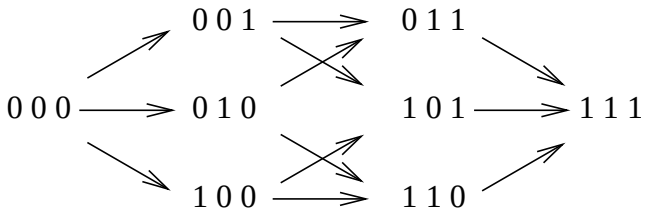


Figure 4.9: Graph of possible binary default states and their transitions for a 3 firm network.

In the same manner as we constructed the default tree in Figure 4.6 and then calculated the expected time for a certain number of defaults to occur, we can calculate the expected time to be in each of the binary states. For our simple example with three firms A, B, and C, 001 refers to a state in which firm C has defaulted and firms A and B have not defaulted. Given the default dependence shown in Figure 4.5, the intensity of the resulting combined Poisson process *once state* 001 *has been reached* is just sum of the intensity of the Poisson process associated with firm A and the intensity of the Poisson process associated with firm B. As firm A does not depend on firm C, there is no change to its default intensity given the default of firm C. However, for firm B the default intensity in increased by ρ_{CB}. Therefore, the intensity of the new combined Poisson process is

$$\lambda_A + \lambda_B + \rho_{CB} \tag{4.50}$$

and hence the expected time to be in state 001 is just

$$\frac{1}{\lambda_A + \lambda_B + \rho_{CB}}. \tag{4.51}$$

We define as $\underline{\phi}$ a vector, ordered by increasing binary values of the bit string, of which each element is the expected time to be in each of the binary states. This is

[2]Remember, AB is the state in which firm A defaulted *and then* firm B defaulted, whereas state BA is where firms B defaulted and then firm A defaulted.

the *expected time vector* and for the example above it is of the form

$$\underline{\phi} = \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \begin{pmatrix} \frac{1}{(\lambda_A+\lambda_B+\lambda_C)} \\ \frac{1}{(\lambda_A+\lambda_B+\rho_{CB})} \\ \frac{1}{(\lambda_A+\lambda_C+\rho_{BC})} \\ \frac{1}{\lambda_A} \\ \frac{1}{(\lambda_B+\lambda_C)} \\ \frac{1}{(\lambda_B+\rho_{CB})} \\ \frac{1}{(\lambda_C+\rho_{BC})} \\ \infty \end{pmatrix}.$$

Note that the expected time to be in state 111 is infinite.

Using the result in (4.43) which gives the probability that an event from a combined Poisson process is associated to a particular one of its underlying processes, we can calculate the probability of moving from one binary state to another *given* that a default event has occurred. This is just the ratio of the default intensity associated with the process of moving from one particular binary state to another binary state over the sum of the intensities of all the processes associated with moving out of that state. This is the same as our calculation of the probabilities associated with each of the states on the default tree. We can then define a probability transition matrix, $\underline{\underline{\Omega}}$, ordered by increasing binary values, where the elements give the probabilities of moving between each of the binary states given a default event. The form if this matrix is shown in Figure 4.10. As one can see from Figure 4.9, it is not possible to move from every state to every other state, and hence many of the elements of $\underline{\underline{\Omega}}$ will be zero to reflect this.

Now we have a vector of the expected time to be in each binary state, $\underline{\phi}$, and a matrix of probabilities of moving from one state to another given a default event, $\underline{\underline{\Omega}}$, we can say that

$$\underline{\mathcal{P}}^d = \sum_{i=0}^{d-1} \underline{\underline{\Omega}}^i . \underline{\phi}, \tag{4.52}$$

where $\underline{\mathcal{P}}^d$ is a vector, of the same form as $\underline{\phi}$, of the expected time for d defaults to occur for every possible initial state from 000 to 111. For example, $\mathcal{P}^2_{001}$ is an element of the vector $\underline{\mathcal{P}}^2$ which gives the expected time for 2 default events to occur given we are in an initial state 001, i.e., an initial state in which C has already defaulted.

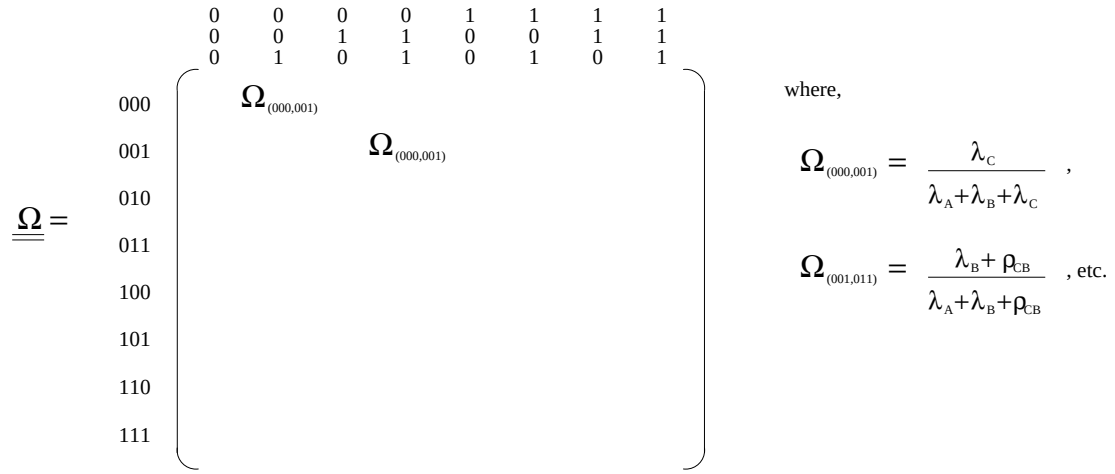


Figure 4.10: Form of the probability transition matrix for a 3 firm network.

When d is greater than the number of undefaulted firms remaining for a given initial state, then the value in $\underline{\mathcal{P}}^d$ corresponding to that state will be infinite.[3]

To calculate the variance of the expected time for a certain number of defaults to occur, one just generates a vector of the same form as the expected time vector, $\underline{\phi}$, but each term is now the square of the each value in $\underline{\phi}$. Note this is analogous to our variance calculation in (4.35) for completely unconnected firms. For our 3 firm example the variance vector is

$$\underline{\chi} = \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \begin{pmatrix} (\frac{1}{(\lambda_A+\lambda_B+\lambda_C)^2} \\ \frac{1}{(\lambda_A+\lambda_B+\rho_{CB})^2} \\ \frac{1}{(\lambda_A+\lambda_C+\rho_{BC})^2} \\ \frac{1}{(\lambda_A)^2} \\ \frac{1}{(\lambda_B+\lambda_C)^2} \\ \frac{1}{(\lambda_B+\rho_{CB})^2} \\ \frac{1}{(\lambda_C+\rho_{BC})^2} \\ \infty \end{pmatrix}.$$

Therefore, the vector of variances, $\underline{\mathcal{V}}^d$, corresponding to the vector $\underline{\mathcal{P}}^d$ of expected

[3]In Appendix B we show a sparse plot of the probability transition matrix for $n_0 = 10$ and give details of the computational implementation of our binary method given above.

times for d defaults for each initial state, is just

$$\underline{\mathcal{Y}}^d = \sum_{i=0}^{d-1} \underline{\underline{\Omega}}^i . \underline{\chi}. \tag{4.53}$$

4.6 Expected Time for a Specific Firm to Default

It is possible using the technique above to calculate the expected time for a specific firm to default for a given initial state. Obviously, it only makes sense to look at initial states where the firm of interest has not yet defaulted.

For our three-firm example, if we are interested in the expected time for firm C to default, we can construct a vector of expected times to be in each state, given by $\underline{\phi}_C$. We set the expected times to be in a state in which C has already defaulted to zero. The probability transition matrix, which gives the probabilities of a move to each state from any other, given a default event has occurred, is also set up as before. However, we set the probability of moving from a state in which C has defaulted to any other as zero.

Hence, we are treating this as a modified process to the general case given above where the final states of the process are ones in which firm C first defaults. Figure 4.11 shows a graph of binary states and their possible transitions for this modified process.

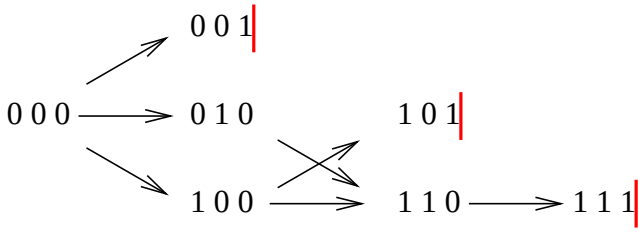


Figure 4.11: Graph of binary default states and possible transitions between states for 3 firms where we are interested in the expected time for firm C to default.

Therefore, we have

$$\underline{\mathcal{P}}_C = \sum_{i=0}^{n_0-1} \underline{\underline{\Omega}}_C^i . \underline{\mathcal{E}}_C, \tag{4.54}$$

where $\underline{\mathcal{P}}_C$ is a vector of the expected times for firm C to default for each initial state, and n_0 is the total number of firms in the network, i.e., in our example $n_0 = 3$.

4.7 Summary of Binary Based General Algorithm Approach

Above we have demonstrated a binary based method to calculate the expected time for any number of defaults to occur and also the expected time for a specific vertice to default, for any general network. By representing the probability of moving between binary states as a matrix and the expected times to be in each binary state as vectors, the approach is useful from both a computational perspective, in terms of ease of implementation and computational costs, and also as a relatively intuitive mathematical framework with which to study our simple contageous default model.

Chapter 5

Analysis of Network Structure

In this chapter we look at the dependence of the expected time to default upon the structure of the network linking the firms together. Initially we look at 3-vertex network structures to develop some general rules which can be applied to larger networks. We then consider more complex network structures and explore the relationship between various network measures and default times. Finally, we apply our model to a specific class of graphs called β-graphs.

5.1 Simple 3-Vertex Networks

Now that we have a method to calculate the expected time to default - under our simple default mechanism, for any combination of firms on any network - we can begin to look at the impact of the network structure itself on the expected default times. Here, we start by considering 3-vertex directed graphs. Of course, even though a real financial network may have many more financial entities than this, building an understanding of the simplest possible structures (aside from the trivial 1 or 2 node examples) will aid the analysis of much more complex networks.

As before, we study 3 firms labelled A, B, and C. In the following examples, to ease our analysis we set all the initial default intensities equal, i.e., $\lambda_A = \lambda_B = \lambda_C = \lambda$, and all non-zero default dependencies equal, i.e., $\rho_{ij} = 0$ or ρ, depending on the network structure. First, we consider all the possible combinations of 3 vertices and 3 edges. As the default intensities and dependencies are all equal, we treat the firms as identical. Therefore, there are only 4 different network structures to compare, i.e., each of these networks are unique except for different permutations of the firms A, B, and C. These networks are labelled blue, green, cyan and red, and their default matrices and graph plots are shown in Figure 5.1.

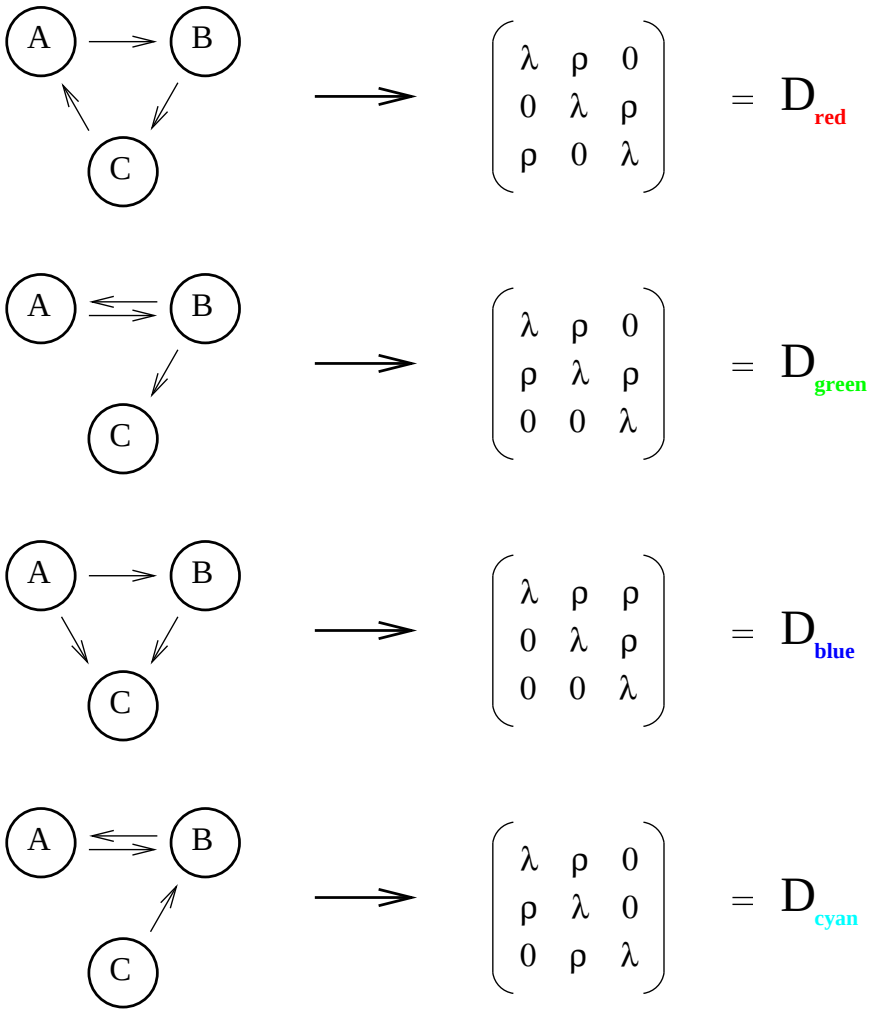


Figure 5.1: Default matrices and graph structures for the 4 possible combinations of 3 vertices and 3 edges.

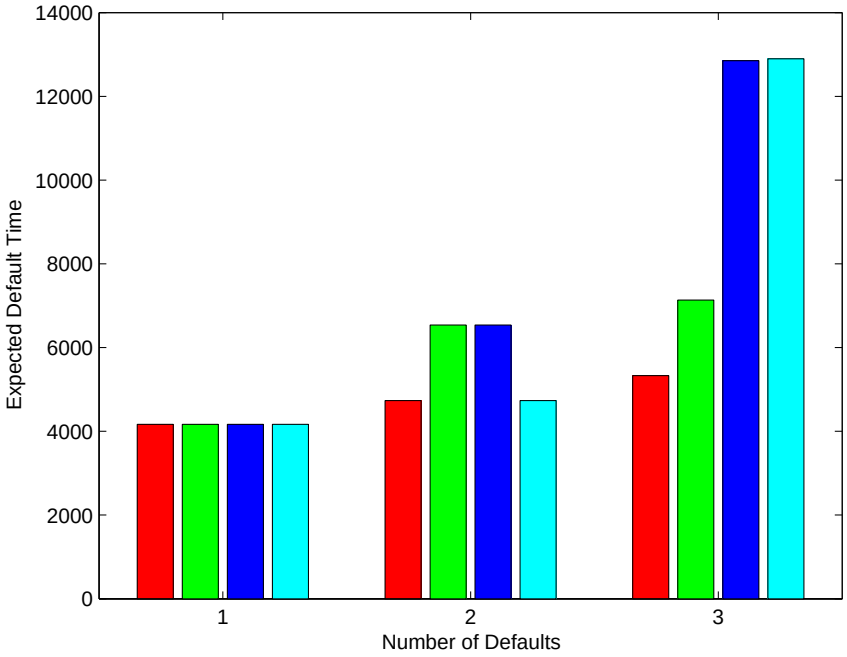


Figure 5.2: Expected time for 1,2 and 3 defaults - for the 4 possible combinations of 3 vertices and 3 edges network structures, with $\lambda = 0.00008$ and $\rho = 0.0016$.

Figure 5.2 shows a plot of the expected time of 1, 2, and 3 defaults to occur for the 4 network structures in Figure 5.1. In this example we have set $\lambda = 0.00008$ and $\rho = 0.0016$. Note, however, that the relative positions of all the expected default times are the same for all $\lambda > 0$ and $\rho > 0$ (as can easily be proved by comparing the expressions for the expected default times given in Appendix C). As one would expect, the expected time for the first default to occur is the same for all structures. This is because the network structure is defined by the default dependencies, which stipulate the change to a firm's default intensity *given* the default of a neighbour. Therefore, for all the networks, the first default is just the expected time of the first event for a combined Poisson process that is the superposition of 3 identical independent Poisson processes.

The expected time for the second firm to default *does*, however,depend on the structure of the network of default dependencies. The red and cyan networks and the blue and green networks have identical expected times for the second default, with the time corresponding to the blue and green networks being greater than that of the red and cyan networks. Explicit proof that this is the case for all $\rho > 0$ where $\lambda > 0$ is given in Appendix C, Section C.5.

Comparing the expected times for all 3 firms to default for the different network

structures, the lowest expected time corresponds to the red network, followed by the green, blue and cyan networks, where the difference between the blue and cyan networks is very small. Hence, the red network appears to be far less stable, or resilient, in terms of the effect upon it of previous defaults, than the cyan, blue and green networks, even though initially they all have the same number of edges.

Rather than calculate explicitly the expected time for each order of default for each network, it is useful to have some other measure by which to compare networks and determine which are more resilient to defaults. This will also provide us with a greater understanding of why two similar networks (same number of vertices and edges) can have such different expected default times.

Before we continue, we introduce some simple terminology from graph theory that is useful in our analysis of the different network structures [31], [32]. Figure 5.3 illustrates the idea of in-degree, out-degree, a source and a sink. Note that for a source, all the entries in the corresponding column in the adjacency matrix for that vertex are zero, whereas for a sink, all entries in the corresponding row in the adjacency matrix for that vertex are zero. A vertex for which all entries in its corresponding row and column in the adjacency matrix are zero is considered neither a source nor a sink as it is unconnected from the rest of the network.

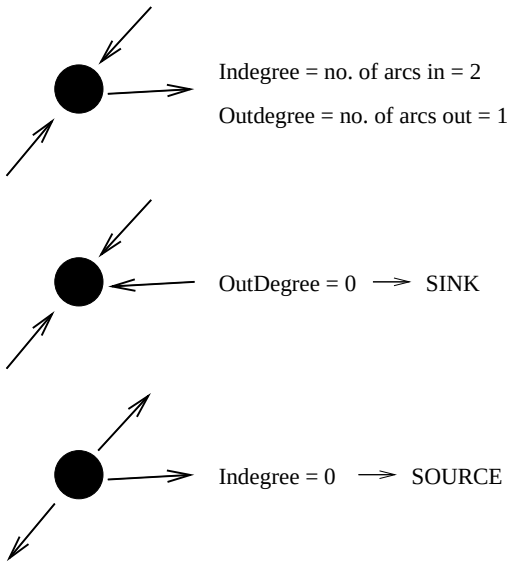


Figure 5.3: Illustration of in-degree, out-degree, a source, and a sink.

Now we introduce the idea of the in-degree and out-degree vectors and use them to compare the 4 network structures above. For an adjacency matrix $\underline{\underline{A}}$, the in-degree

vector $\underline{\nu}_{in}$, and out-degree vector $\underline{\nu}_{out}$ are defined as

$$\nu_{in}(i) = \sum_{j=1}^{n_0} A_{i,j}, \tag{5.1}$$

$$\nu_{out}(i) = \sum_{j=1}^{n_0} A^T_{i,j}, \tag{5.2}$$

where $\underline{\underline{A}}^T$ is the transpose of $\underline{\underline{A}}$ and n_0 is the total number of vertices - i.e., the in-degree vector is the column sum, and the out-degree vector is the row sum.

For each of the 4 networks, the in-degree and out-degree vectors are shown in figure 5.4.

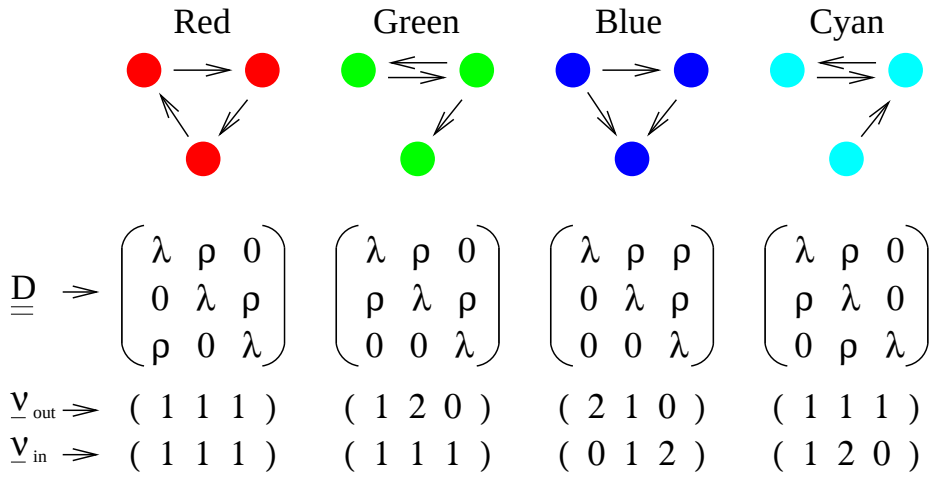


Figure 5.4: In-degree and out-degree vectors for the 4 network structures for 3 vertices and 3 edges.

As the expected time for the first default to occur is the same for all the networks, we move straight away to look at the expected times for the second default to occur. If one compares the out-degree vectors for the four networks in Figure 5.4, where a zero value or sink exists, then one sees that the expected time for the second default is greater for these networks as compared to the networks where no sinks are present. The presence of a vertex that is a sink is significant as it will have no effect on the other vertices (or firms) in the network were it to be the first to default. Where a sink exists in the out-degree vectors of Figure 5.4, then there must also be another vertex with an out-degree of 2. Were this vertex with out-degree 2 to default first, then 2 neighbouring vertices would have increased default intensities. However, the reduction in the expected time of the second default, of having a vertex with out-

degree of 2, is not enough to counteract the increase in the expected time due to the presence of a sink.

Therefore, for the 3-vertex, 3-edge networks under consideration, the greater the number of sinks or zeros in the out-degree vector, the greater the expected time of the occurrence of the second default.

Now we look at the expected times of the third default to occur. If one makes a comparison of the expected default times for the red and cyan networks in Figure 5.2, one can see that the default time for the cyan network is greater than that of the red network, even though they have the same expected times for the second default. By comparing the in-degree vectors for both networks, one can attribute this difference in expected default times to the presence of a source vertex in the network, given that both networks have identical out-degree vectors. One can also see that this is the case for the green and blue networks.

Finally, as the expected time for the third default for the cyan network is greater than that for the green network, then this implies that the impact of a source in the in-degree vector is greater that the impact of the sink in the out-degree vector.

Now we consider the 4 possible network structure combinations of 3 vertices and 2 edges shown in Figure 5.5, with their associated in-degree and out-degree vectors. Again we have labelled the 4 structures as red, green, blue, and cyan.

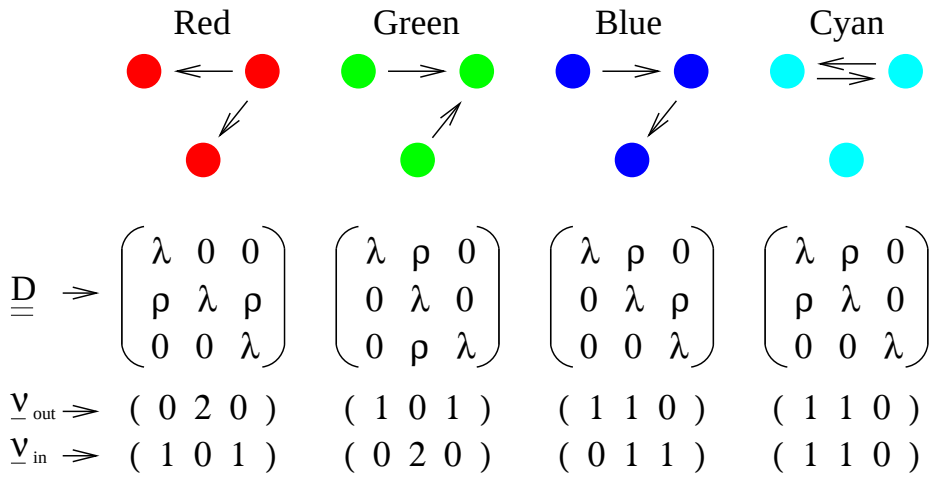


Figure 5.5: In-degree and out-degree vectors for the 4 network structures for 3 vertices and 2 edges.

Figure 5.6 shows the default times for the 4 network structures, with $\lambda = 0.00008$ and $\rho = 0.0016$, as before. The first observation is that the relative difference between the default times is less marked than for the 3 vertex 3 edge networks in Figure 5.6.

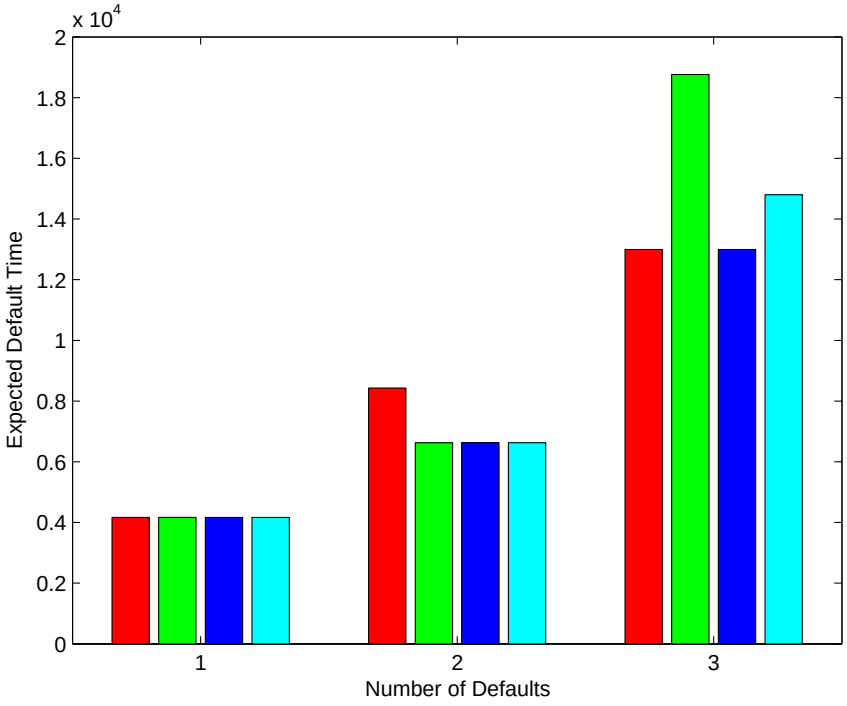


Figure 5.6: Expected time for 1,2 and 3 defaults - for the 4 possible combinations of 3 vertices and 2 edges network structures, with $\lambda = 0.00008$ and $\rho = 0.0016$.

Again, notice that the out-degree vector is a measure of the relative position of the expected times of the *second* default event. The presence of the 2 sinks in the red network increases the expected default time over that of the green, cyan, and blue networks, which all have only one sink.

Given that the green, cyan and blue networks have the same expected times for the second event, the presence of 2 sources (indicated by 2 zero elements in the in-degree vector) in the green network means it has a greater expected default time than the cyan and blue networks, which only have one source. Notice also that the presence of 2 sinks in the red network is not significant as the presence of 2 sources in the green network when comparing the total expected time for all defaults to occur. We saw this pattern earlier with the cyan and green 3-vertex 3-edge networks.

The reason for the difference in the expected times of the third default for the cyan and blue networks, given that they both have the same number of sinks and sources, may lie in the presence of a cycle in the cyan network and no cycles in the blue network, and hence because there are only 2 edges in the network, the presence of an unconnected vertex in the cyan network, see figure 5.7.

By considering both the 3-vertex, 3-edge networks and the 3-vertex, 2-edge net-

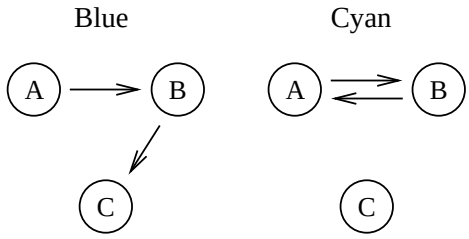


Figure 5.7: Structures of green and blue 3 vertex, 2 edge networks.

works, we have shown that in this instance the out-degree vector can be used as some measure of the relative expected times of the second default for the different structures (with the same number of edges). Also, the in-degree vector can help predict the relative values of the expected times for all 3 firms to default, given identical expected times for the second default event. Finally, one may compare the overall expected times for all firms by noting that the impact of a source in these networks is greater than that of a sink. However, even with the simple 3 vertex structures analysed above, there are other factors such as the presence of cycles and unconnected vertices that can further differentiate the networks in terms of their associated expected times to default. Therefore, the formulation of a simple yet general rule for translating a network's adjacency matrix to its resilience-to-default events, without explicitly calculating the expected times, has not proved possible. Nevertheless, for application to general networks of all sizes and complexity, our study into the 3 vertex structures has highlighted some determinants of network resilience, such as the dominant effect upon the behaviour of these networks of sources, sinks, and cycles.

In the next section we show that for larger 4 and 5 vertex networks, only very small changes to the network structure, such as the movement of one edge, can have dramatic effects on the expected time to default. The results also highlight again the importance of sinks, sources, and especially cycles on the expected times to default, and show that other measures of network structure are needed to predict the response a network to defaults.

5.2 Expected Default Times for some 4 and 5 Vertex Networks

In the following examples, we highlight the difference to the expected default times of small changes to the structure of some 4 and 5 vertex networks. Figures 5.8, 5.9, and 5.10 show 3 pairs of networks and their associated expected default times. The only

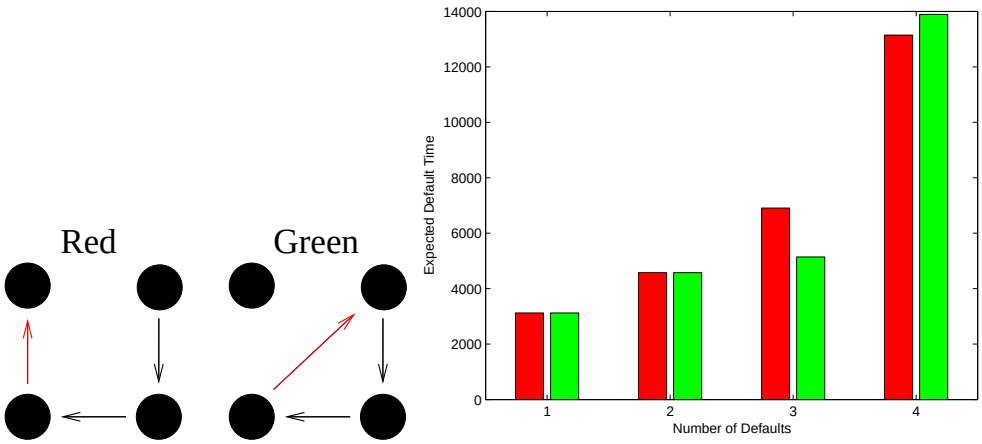


Figure 5.8: Expected time for 1 to 4 defaults for the two different 4 vertex, 3 edge networks illustrated, with $\lambda = 0.00008$ and $\rho = 0.0016$.

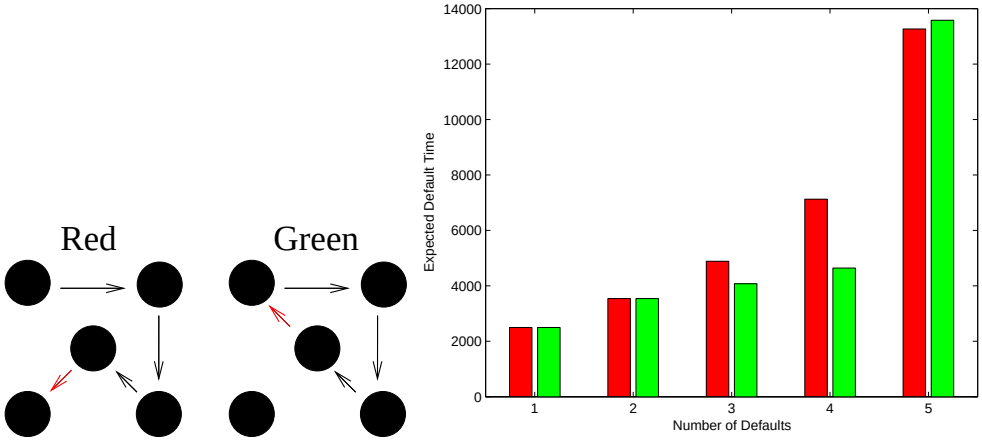


Figure 5.9: Expected time for 1 to 5 defaults for the two different 5 vertex 4 edge networks illustrated, with $\lambda = 0.00008$ and $\rho = 0.0016$.

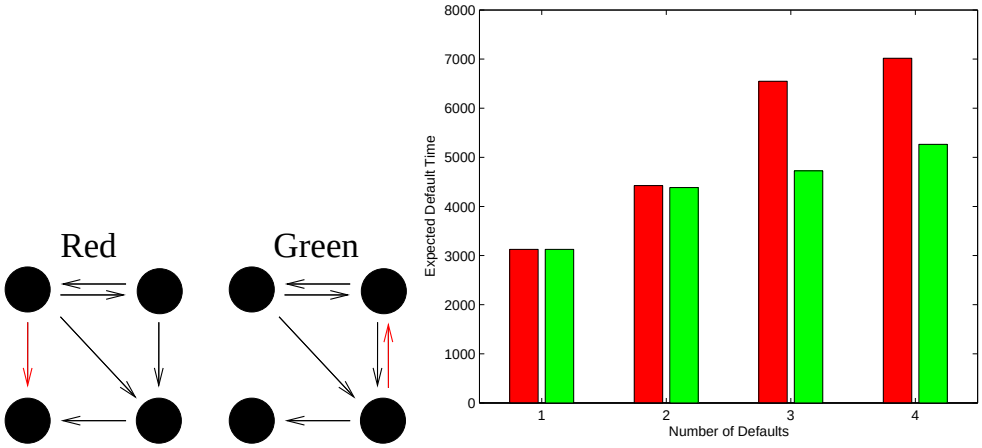


Figure 5.10: Expected time for 1 to 4 defaults for the two different 4 vertex 6 edge networks illustrated with $\lambda = 0.00008$ and $\rho = 0.0016$.

difference between each network pair is the repositioning of one edge, highlighted in red. The network pairs in Figures 5.8 and 5.9 are similar in that the red networks are fully connected whilst the green networks are not fully connected, with the movement of one edge forming a cycle. The relative expected default times between the red and green networks in Figures 5.8 and 5.9 show that even though the green networks are more resilient to *all* of their vertices defaulting, they have lesser expected default times for the second default up to the $(n_0 - 1)$th default. Therefore, in the measure of network resilience, one must take account of what fraction of the total number of vertices, n_0, that are under consideration. Clearly here, the presence of the cycle and the unconnected vertex has a significant effect on network resilience but in both a positive and negative way depending upon how many defaults occur. This is relevant to the value of an nth to default basket, where n is the number of firms in the basket to default. For the above networks, one would prefer to purchase a contract written on a basket of firms governed by the dependence illustrated by the green network (all else being equal), if one received some payoff given the default of all the firms. A basket of firms whose dependence was given by the red network, however, would be a better underlying if the payoff was made given $n_0 - 1$ defaults.

Figure 5.10 shows the strong impact on the expected default times of repositioning just one vertex in a 4-vertex, 6-edge network. The large difference is perhaps surprising given that both the green and red networks still have 5 edges in common. Both the red and green networks contain cycles, a source, and a sink. Certainly, it is not easy to predict the network's relative responses to default events from a visual assessment of the two network structures alone. Hence, for larger networks other measures of their structure are required to understand the behaviour of the networks when defaults occur.

5.3 Measures of Default Dependence in Complex Network Structures

In this section we consider various measures of complex networks and ascertain which are useful in predicting the response of a given network structure to the process of its nodes successively defaulting. The measures we consider here have been widely used in the analysis of complex network structures, especially the subset of complex networks referred to as small-world [46]. As will become apparent, these measures allow us to broadly describe, in a macroscopic sense, the structure of a network, with-

out distinguishing exactly the microscopic nature of the placement of edges between vertices, i.e., without stipulating the exact form of the adjacency matrix $\underline{\underline{A(t)}}$.

First we construct a random directed graph of a given number of vertices and edges, and given values of the individual initial default intensity, λ, and the default dependency, ρ, where all values of λ and ρ are the same, we then calculate the expected time to default of a particular number of the vertices. On this same structure we calculate the network measure under consideration. This process is then repeated many times to ascertain if a relationship exists between the expected default times and the network structure measure.[1]

For larger network structures (approximately more than 15 vertices) we used a Monte Carlo simulation to calculate the expected time to default rather than the tree based approach detailed in the previous sections. This is because of the computational time required by the tree based approach when dealing with large numbers of vertices. (In Appendix B we mention in more detail the computational implementation of the tree based algorithm.) The Monte Carlo method of calculating the expected time to default is straightforward. At each iteration of the simulation, cycling though each vertex in turn, a random number between 0 and 1 is drawn from a uniform distribution, and compared to the default intensity of the firm represented by that vertex. If the random number is less than or equal to the default intensity then that firm is deemed to have defaulted. Once each vertex has been acted on, the default intensities of each of the nodes are then adjusted to account for any defaults of neighbouring vertices. The simulation runs until the required number of vertices has defaulted and the number of iterations is recorded. The whole process is then repeated many times and an average default time is then calculated.

For a directed graph of n_0 vertices, the total number of unique edges that can be placed between on that graph is $n_0(n_0-1)$, i.e., each vertex can have n_0-1 directed edges leaving it to each of the other vertices. Therefore, the number of possible network realisations that can be created given n_0 vertices and L edges for a directed graph is

$$\binom{n_0(n_0-1)}{L}. \tag{5.3}$$

For identical vertices and identical edge weights each of these realisations are only unique up to permutation. However, in our simulation, each of the possible network

[1] In our initial study we looked at a range of parameters including number of vertices, number of edges, and number of network realisations. The results we present are using values for those parameters that we consider to give some of the strongest dependencies and clearest results.

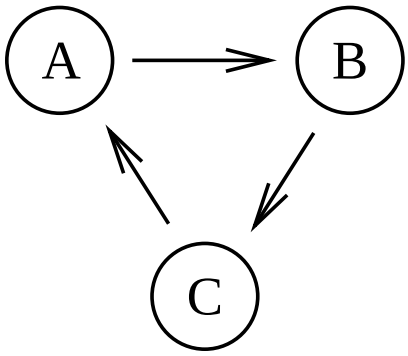


Figure 5.11: Illustration of cycles of length 3. There are 3 cycles here, $A \to B \to C$, $B \to C \to A$, and $C \to A \to B$.

realisations can be generated with equal probability. For 7 vertices and 12 edges the number of possible graphs is $\sim 10^9$. Therefore, we acknowledge that our samples of perhaps 1000 realisations is small compared to such a large population. However, the nature of this study is to look for general trending rather than exact statistical descriptions of the networks.

As was shown in the previous sections, one network may be more or less resilient than another depending upon the number of defaults we are considering. However, for our analysis, we decided that a fair measure of a network's resilience to default events is the expected time for approximately half of its vertices to default. The justification for this is that for our default mechanism, we have shown that for any network the expected time of the first default is the same - as the effect of the network dependence is only realised after defaults have occurred. Therefore, the true nature of the default dependence can only be determined once the vertices begin to default. However, from a financial market perspective, if we looked at the expected times for all of the vertices (firms) to default, then a situation where this occurred (unless the firms are incredibly high risk initially) is probably very unlikely. Considering some number of defaults between these two extremes is therefore a compromise between allowing some defaults to occur but not so many as to perhaps be considered unrealistic from a real world perspective.

5.3.1 Number of Small Cycles

The first measure we consider is the number of cycles of length 3. A path is a sequence of edges that begins at an initial vertex and ends at a terminal vertex. A cycle is a path that begins and terminates at the same vertex. A directed cycle of length 3 is

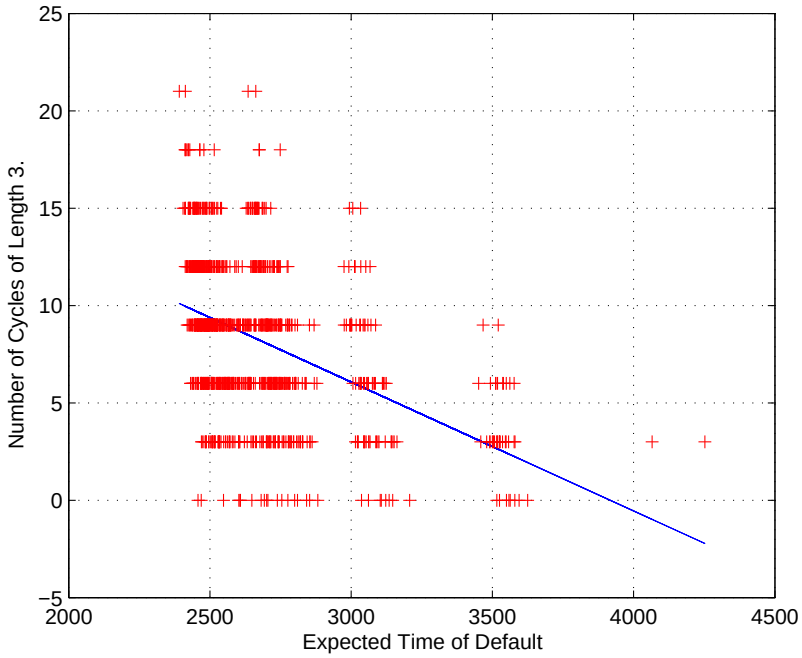


Figure 5.12: Plot of the number of cycles of length 3 for a directed graph with 7 vertices, 15 edges versus expected time for 4 defaults; $\lambda = 0.00008$, $\rho = 0.0016$, 1000 network realisations. The blue line is a linear fit to the data.

shown in Figure 5.11. In fact there are 3 directed cycles in the diagram; $A \to B \to C$, $B \to C \to A$, and $C \to A \to B$. For a directed graph with adjacency matrix $\underline{\underline{A}}$, the calculation of the number of cycles of length w is just

$$\text{Trace}(\underline{\underline{A}}^w). \tag{5.4}$$

Figure 5.12 shows a plot of the number of cycles of length 3 versus the expected time for 4 defaults to occur in a 7-vertex, 15-edge directed graph. Note that the number of cycles are all multiples of three as we should expected from our illustration of a cycle in figure 5.11. Whilst there does appear to be some relationship - an increase in the number of cycles means a decreased expected time - it is not particularly strong. However, this relationship is what one might expect, given the analysis of the 3, 4 and 5-vertex networks. There we saw a reduction in the expected time of the default of some fraction of the vertices if cycles were present in that network, compared to a similar network with no cycles. Therefore, this is provides more evidence of the impact of cycles on the expected time to default.

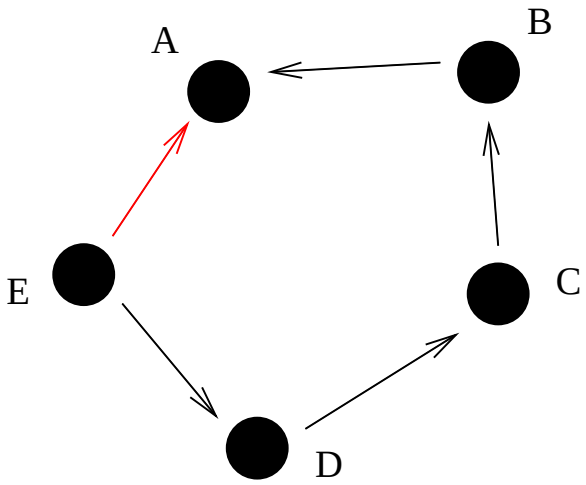


Figure 5.13: Illustration of a shortcut.

5.3.2 Range and Shortcuts

Here we consider whether measuring the fraction of edges in the network that are shortcuts is a good predictor of the resilience of a network to default events.

The range, $R(i, j)$, of an edge is defined as the shortest path between two vertices i and j in the absence of that edge, i.e. $R(i, j)$ is the second shortest path length between i and j [46]. The path length is defined such that $+1$ is added to the length for each edge traversed on that path. For a directed network we distinguish between $R(i, j)$ and $R(j, i)$. If an edge has range greater than 2 then it is considered to be a shortcut.[2] This idea is illustrated in Figure 5.13. The red directed edge linking vertex E to vertex A is a shortcut, because in its absence, the shortest path running from E to A is of length 4.

Figure 5.14 is a plot of the fraction of edges in a directed network that are shortcuts versus the expected time for 4 defaults, for a 7 vertex, 11 edge network structure. There does appear to be some relationship; an increase in the fraction of shortcuts leads to a decrease in the expected default time, although this is far from adequate as a good predictor of network resilience. In fact, as the figure shows, some of the network realisations with zero edges that are shortcuts have the same expected time for 4 of its vertices to default, and hence the same resilience to defaults (as we have defined it), as network realisations in which *all* the edges are shortcuts.

[2]Note, if a directed edge is the only link in a particular direction between two vertices, then the range of the edge is effectively infinite. However, for our analysis, we do not consider this edge to be a shortcut even though the range is greater than 2.

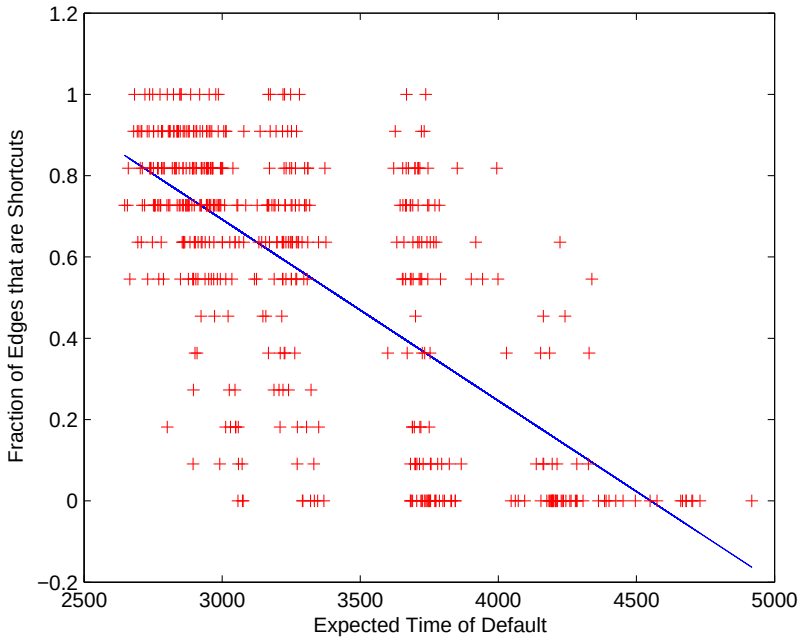


Figure 5.14: Plot of the fraction of edges that are shortcuts versus expected time for 4 defaults. 7 vertices, 11 edges, $\lambda = 0.00008$, $\rho = 0.0016$, 1000 network realisations. The blue line is a linear fit to the data.

5.3.3 Edge Complexity

The Watts edge complexity of a network is the *average deviation of the range, R, over all edges in the network* [46]. Again, as with the measure of shortcuts, for a directed network we distinguish between the range between vertex i and j, given by $R(i, j)$, and the range between vertex j and i, given by $R(j, i)$. In general, the average deviation is given by,

$$s = \frac{1}{n} \sum_{i=1}^{n} |x_i - \bar{x}|, \tag{5.5}$$

where n is the number of elements in the set, x_i is some measure of element i, and $\bar{x}$ is the mean of that measure over all elements in the set. Note that this measure is less sensitive to outliers than the more commonly used standard deviation as the average deviation takes the absolute value of deviations instead of the square of deviations from the mean. In the context of the edge complexity measure, an outlier refers to an edge with a particularly large range such that it might have a disproportionate affect on the standard deviation.

For directed edges that are the only link between two vertices in a given direction, the range of that edge is defined to be infinite. Therefore, we have chosen to consider

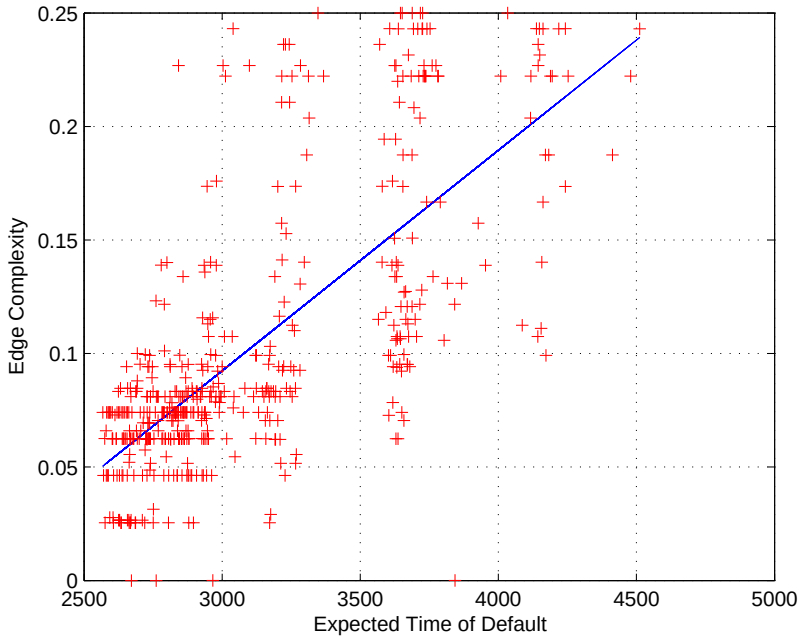


Figure 5.15: Plot of the edge complexity versus expected time for 4 defaults. 7 vertices, 12 edges, $\lambda = 0.00008$, $\rho = 0.0016$, 500 network realisations. The blue line is a linear fit to the data.

the average deviation of the reciprocal of the range from the mean of the reciprocal of the range. This means that an infinite range will add nothing to the sum of the reciprocals of the ranges.

The idea behind the Watts edge complexity is to build some measure of the *statistical complexity* of a network. Statistical complexity is a statistic put forward by Crutchfield who states that whilst complexity is a purely subjective measure, the measure of the complexity of a system can be approached in terms of the information required to optimally forecast the outcome of that system [33]. Therefore, a system that is purely random is not complex, as the future output of a completely stochastic system may be statistically described by possibly just one parameter, or a 1-parameter family of probability density functions. Also, a completely periodic signal is not complex as it can be described by only one parameter. Hence, a complex system, defined here by the information needed to describe it, lies somewhere between a completely random construction and a completely ordered structure. This measure of complexity Crutchfield terms *statistical complexity*.

Watts argues that a complex network can also be considered in terms of the information required to describe it. Hence, a complex network is one that lies somewhere between a completely random network, where each edge is placed at random, and a

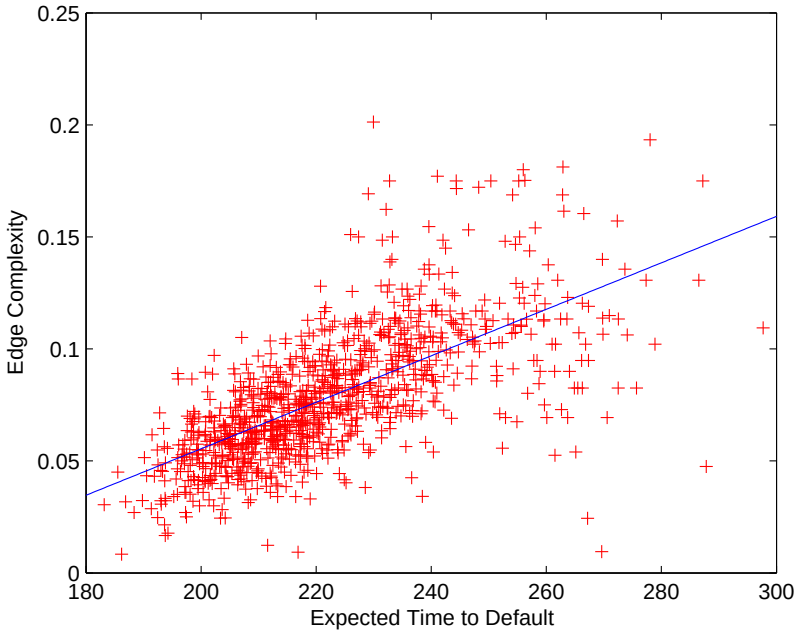


Figure 5.16: Plot of the edge complexity versus expected time for 12 defaults for 1000 network realisations of 25 vertices and 40 edges, with $\lambda = 0.0008$ and $\rho = 0.016$. Monte Carlo averaging of expected time to default for each network structure realisation over 1000 runs. The blue line is a linear fit to the data.

regular network structure such as a lattice. He postulates that a network can therefore be considered complex when a small fraction of its edges are long-range shortcuts - and that this can be measured by the average deviation of the range of all edges in the network - the edge complexity.

Figure 5.15 is a plot of the edge complexity versus the expected time for 4 defaults for a 7 vertex, 12 edge network structure. As one can see, very approximately, as the edge complexity increases so does the expected time of default and hence, the resilience of the structure. This trend is also evident for an analysis of 25 vertex, 40 edge network structures, shown in Figure 5.16, where Monte Carlo averaging has been used to determine the expected time to default. Therefore, we can conclude that whilst the edge complexity may be considered a rough guide to the resilience of the network, given the large range of edge complexity values for approximately the same expected time of default, it is a long way from being an accurate predictor of the expected time for half the firms in the network to default.

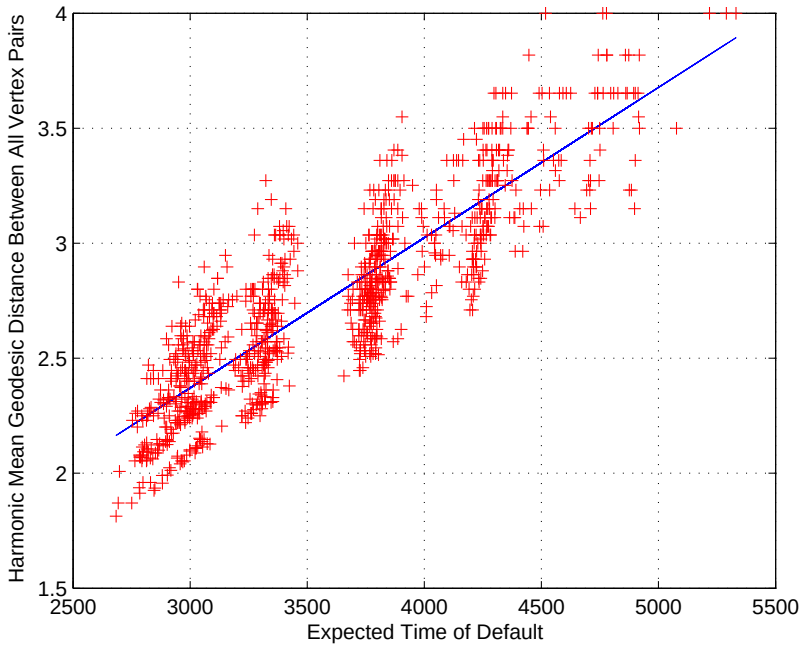


Figure 5.17: Plot of the harmonic mean geodesic distance between all vertex pairs versus expected time for 4 defaults. 1000 network realisations studied, consisting of 7 vertices, 10 edges, with $\lambda = 0.00008$ and $\rho = 0.0016$. The blue line is a linear fit to the data.

5.3.4 Average Path Length

Here we consider the impact of the average path length of the network on the resilience of the network structure to defaults. More precisely we look at the harmonic mean geodesic distance between all pairs of vertices in the directed graph, where we distinguish between the distance between i to j and j to i.

It is possible that for some realisations of the network that some of the vertices are not connected to any other, or that the network is split into separate groups, i.e. the network is not *complete*. That is why we use the harmonic mean geodesic distance, l, to measure the path length between all vertex pairs [47]. This is given by:

$$l^{-1} = \frac{1}{\frac{1}{2}n_0(n_0+1)} \sum_{i \geq j} d_{ij}^{-1}, \tag{5.6}$$

where i and j are any two vertices, n_0 is the total number of vertices and d_{ij} is the geodesic distance from vertex i to vertex j. Using the harmonic mean means that infinite values of d_{ij} between unconnected vertices contributes nothing to the sum.

As one can see from Figure 5.17, there is a strong relationship between the harmonic mean geodesic distance between all vertex pairs and the expected time of the

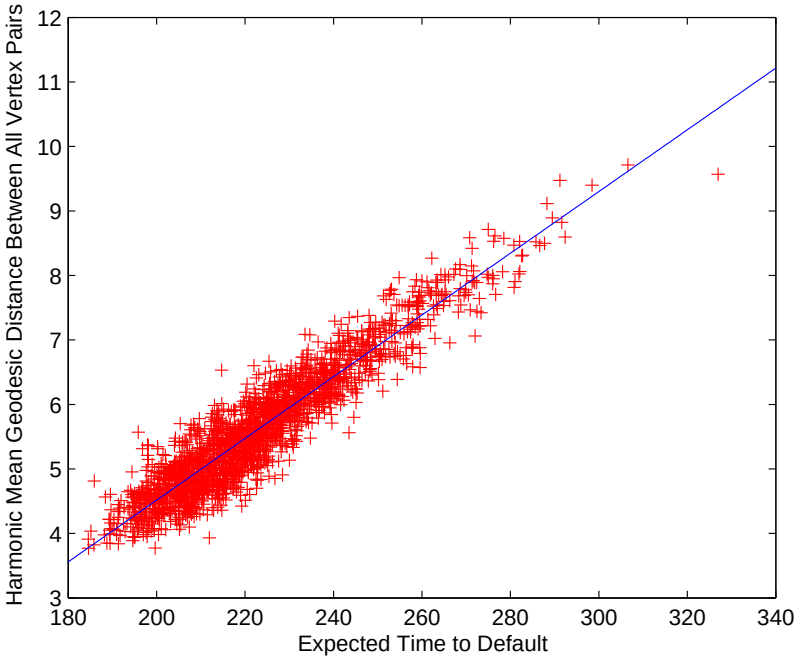


Figure 5.18: Plot of the harmonic mean geodesic distance between all vertex pairs versus expected time for 12 defaults for network realisations of 25 vertices and 40 edges with $\lambda = 0.0008$ and $\rho = 0.016$. Monte Carlo averaging of expected time to default for each network structure realisation over 1000 runs. The blue line is a linear fit to the data.

fourth default for a 7-vertex, 10-edge network structure. As the average path length between the vertex pair increases, so does the resilience of the network to default events. This may be expected, as the "further apart", in terms of path length, that two vertices are, the greater the number of default events that will have to occur before the default of one vertex has an impact on the other. For a small average path length, all vertices on average will have a fairly direct impact, given default, on all the other vertices, i.e., the network is highly connected, and, hence, the network is not resilient to default events occurring.

Aside from the general trend mentioned above, there also is fairly clustering in the expected default times, which appears in the figure as vertical bands. This also appears to be present in the other lower vertex simulations such as in Figure 5.15 for the edge complexity. We do not address this here although from our study it does appear to be dependent on the number of vertices, i.e., it is not present in Figure 5.16 and Figure 5.18.

Figure 5.18 also shows a strong relationship between the harmonic mean geodesic distance and the expected time to default, where here we are concerned with the 12th default on a 25-vertex, 40-edge network. Again, rather than use the tree based algorithm for calculating the expected time to default, for a network of this size we have used Monte Carlo averaging of the default mechanism.[3]

From both plots, it is clear that given two different structures with very different average path lengths, it would be possible to say with some confidence that one is more resilient than the other. In fact, the average length appears to be an even better predictor of resilience when one considers larger, more complex network structures. Therefore, we can us the harmonic mean geodesic distance as some indicator of network stability without having to make a detailed analysis of the exact structure of the network.

5.4 Application of Default Mechanism to β-graphs

In this section we study the application of our simple additive default mechanism to a class of network called β-graphs [46]. The algorithm behind the construction of a β-graph is to first create a regular lattice structure, such as the 1-lattice shown in Figure 5.19.[4] Then, moving through each vertex, i, in turn, take the edge that connects this vertex to its nearest neighbour in a clockwise direction, and rewire this

[3] In appendix E we discuss the accuracy of the tree method compared to Monte Carlo averaging.

[4] In a 1-lattice or one dimensional lattice, each vertex has 2 nearest neighbours, 2 next nearest neighbours and so on.

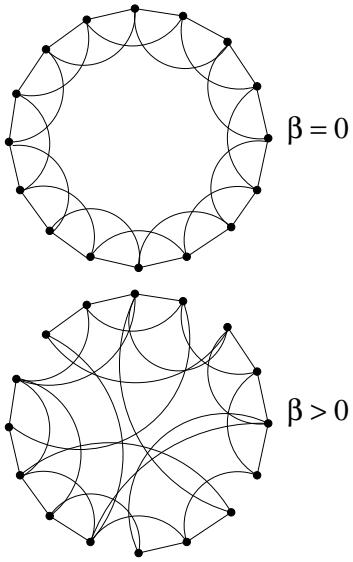


Figure 5.19: Illustration of the concept of the β-graph.

edge with a probability β to another vertex j, where j is chosen uniformly at random and $j \neq i$. Once the clockwise nearest neighbour of each vertex has been considered, move through each vertex again, this time re-wiring each clockwise second nearest neighbour with a probability β. Repeat this pattern until all edges in the network have been considered for re-wiring exactly once.

Figure 5.20 shows the expected times to default for each vertex for a 100 vertex β graph with 8 undirected edges per vertex for a range of β values between 0 and 1. By undirected we mean that each edge represents an equal and opposite dependence between each vertex. Therefore, each undirected edge is essentially 2 cycles of length 2, one cycle for each vertex. As one can see, as β is increased, the value of the expected time to default is decreased, especially for defaults 50 and above. In fact, the higher the number of defaults under consideration, the greater the difference in default time for different values of β. The effect of changing β for a given default order (e.g., the 75th default) is not linear and decreases with increasing β.

From these results one would infer that for our simple default mechanism, the more random the network structure, the less resilient the network is to defaults. One might expect that the more edges rewired, the wider the influence across the whole network that each default has and hence the less stable the network is to default. However, it is interesting to note that if one considers the statistical complexity of the networks with increasing β (see Section 5.3.3), then the most complex networks lie somewhere between $\beta = 0$ and $\beta = 1$, in terms of the information required to

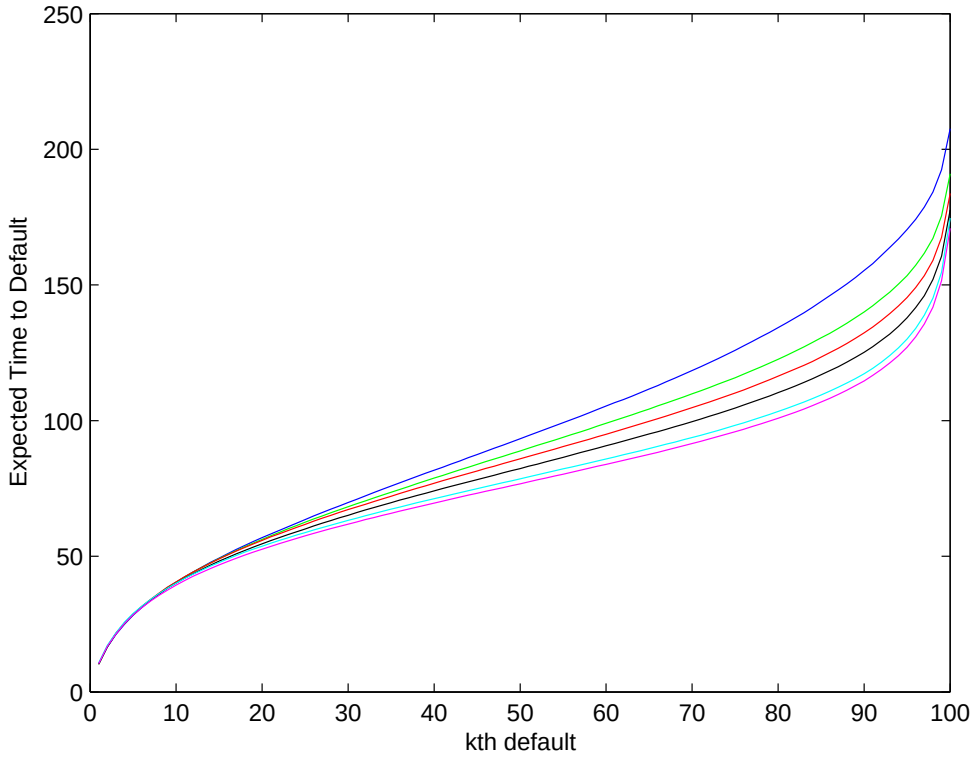


Figure 5.20: Plot of expected time to default versus order of default for 100 vertex beta graph with 8 undirected edges per vertex. Expected time calculated by Monte Carlo averaging default process over 2500 runs. Blue line represents $\beta = 0.0$, green line $\beta = 0.05$, red line $\beta = 0.1$, black line $\beta = 0.2$, cyan line $\beta = 0.5$, magenta line $\beta = 1.0$.

describe the network. Therefore, here, an increase in the complexity of the network does not necessarily lead to a decrease in the resilience of the network structure to defaults.

Chapter 6

Pricing Baskets and Bonds

In this chapter we build on the tree-based method of calculating the expected time to default given in Chapter 4. We detail how to compute the probability of being in a particular state on the tree and then use this to price a simple hybrid kth to default basket.

6.1 Probability of Being in a Particular State on Default Tree

Consider the very simple 2 firm default network illustrated in Figure 6.1, with default matrix $\underline{\underline{D}}$ given by

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda_A & 0 \\ \rho & \lambda_B \end{pmatrix},$$

where for simplicity we have set $\lambda_A = \lambda_B$. Using the same method as in Section 4.4, we can construct a very simple default tree of the possible states that can occur as this network defaults along with the associated probabilities of, given a default event,

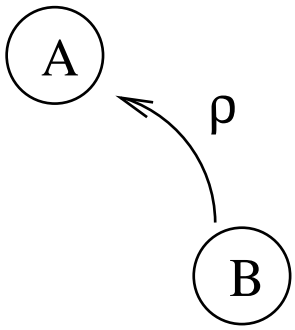


Figure 6.1: Simple 2 firm network.

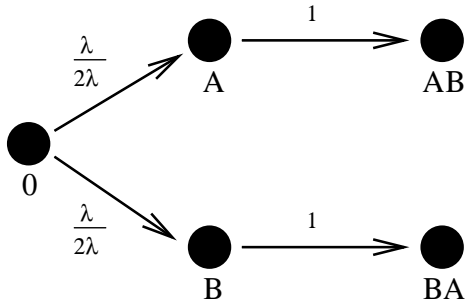


Figure 6.2: Possible states of simple 2 firm network.

moving to another state. This is shown in Figure 6.2, where, for example, AB is the state in which A has defaulted and then B defaults.

Let us first consider state 0 in which no defaults have occurred. Let $P_0(t)$ be the probability of being in state 0 at time t. We can then write down the probability of being in state 0 at $(t + dt)$ as

$$P_0(t + dt) = P_0(t)(1 - 2\lambda dt), \tag{6.1}$$

where $(1-2\lambda dt)$ is the probability if not leaving state 0 in time dt, i.e., the probability of not defaulting. Using a Taylor expansion we find that

$$\begin{aligned} P_0(t) + \frac{dP_0(t)}{dt}dt &= P_0(t) - 2\lambda P_0(t)dt & (6.2)\\ \frac{dP_0(t)}{dt}dt &= -2\lambda P_0(t). & (6.3) \end{aligned}$$

Thus, using the initial condition $P_0(0) = 1$ we have

$$P_0(t) = e^{-2\lambda t}. \tag{6.4}$$

Now we have an expression for the probability of being in state 0 at time t we can move along the tree in Figure 6.2 and write down an expression for the probability of being in state A at time t,

$$\begin{aligned} P_A(t + dt) &= P_0(t)\Big(\frac{\lambda}{2\lambda}\Big)(2\lambda dt) + P_A(t)(1 - \lambda dt) & (6.5)\\ P_A(t + dt) &= P_0(t)(\lambda dt) + P_A(t)(1 - \lambda dt), & (6.6) \end{aligned}$$

where $2\lambda dt$ is the probability of leaving state 0 and moving into state either state A or state B, $\frac{\lambda}{2\lambda}$ is the probability of moving into state A *given* a move out of state 0 and $(1 - \lambda dt)$ is the probability of not leaving state A in time dt.

Expanding in a Taylor series, we find that

$$\frac{dP_A(t)}{dt} = \lambda P_0(t) - \lambda P_A(t). \tag{6.7}$$

As we know already that $P_0(t) = e^{-2\lambda t}$, we now have

$$\frac{dP_A(t)}{dt} = e^{-2\lambda t} - \lambda P_A(t), \tag{6.8}$$

with $P_A(0) = 0$ as, initially, the system is in state 0. We can solve this using the integrating factor $e^{\lambda t}$ to get

$$P_A(t) = e^{-\lambda t} - e^{-2\lambda t}. \tag{6.9}$$

Similarly, for state B we have

$$P_B(t + dt) \;=\; P_0(t)(\lambda dt) + P_B(t)(1 - (\lambda + \rho)dt), \tag{6.10}$$

where $(1 - (\lambda + \rho)$ is the probability of not leaving state B in time dt. Note, the ρ is the default dependence on A of B defaulting. Again, using a Taylor expansion we find

$$\frac{dP_B(t)}{dt} = \lambda P_0(t) - (\lambda + \rho)P_B(t), \tag{6.11}$$

with $P_B(0) = 0$ so that

$$P_B(t) = \frac{\lambda}{(\rho - \lambda)}\Big[e^{-2\lambda t} - e^{-(\lambda+\rho)t}\Big]. \tag{6.12}$$

To summarize, we have,

$$P_0(t) \;=\; e^{-2\lambda t} \tag{6.13}$$

$$P_A(t) \;=\; e^{-\lambda t} - e^{-2\lambda t} \tag{6.14}$$

$$P_B(t) \;=\; \frac{\lambda}{(\rho - \lambda)}\Big[e^{-2\lambda t} - e^{-(\lambda+\rho)t}\Big]. \tag{6.15}$$

We are now in a position to calculate the probability of two defaults occurring between time $t = 0$ and $t = T$. The probability of being in state AB during the time period $(t, t + dt)$ is

$$P_A(t)\lambda dt, \tag{6.16}$$

where $P_A(t)$ is the probability of being in state A at time t, and λdt is the probability B defaults in time dt. Similarly, the probability of being in state BA during the time period $(t, t + dt)$ is given by

$$P_B(t)(\lambda + \rho)dt. \tag{6.17}$$

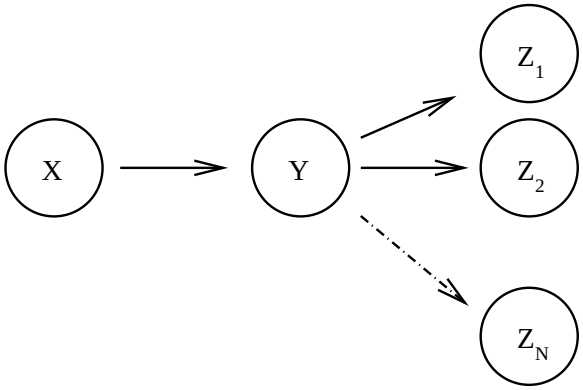


Figure 6.3: Illustration of general case default tree with one vertex being fed by one other, and itself feeding N others.

Hence, the probability of two defaults occurring (i.e., reaching either state AB or state BA) between $t = 0$ and $t = T$, is just given by

$$\int_0^T (P_A(t)\lambda + P_B(\lambda + \rho))dt, \tag{6.18}$$

where, by substituting in for $P_A(t)$ and $P_B(t)$ gives us

$$\int_0^T (\lambda e^{-\lambda} - \lambda e^{-2\lambda t} + \frac{(\lambda + \rho)}{(\rho - \lambda)}\lambda e^{-2\lambda t} - \frac{(\lambda + \rho)}{\rho - \lambda)}\lambda e^{-(\lambda+\rho)t})dt, \tag{6.19}$$

and then solving gives

$$-e^{\lambda T} + \frac{1}{2}(1 + e^{-2\lambda T}) + \frac{(\lambda + \rho)}{2(\rho - \lambda)}(1 - e^{-2\lambda T}) + \frac{\lambda}{(\rho - \lambda)}(e^{-(\lambda+\rho)T} - 1). \tag{6.20}$$

6.1.1 General Method

In general, for any default tree, the probability of being in state Y at time t, given that state Y is fed by state X, and that state Y feeds states (Z_1), (Z_2), ... (Z_N) (see figure 6.3) is given by,

$$\frac{dP_Y(t)}{dt} = \lambda_{X\to Y} P_X(t) - \Big[\sum_{i=1}^{N} \lambda_{Y\to Z_i}\Big] P_Y(t), \tag{6.21}$$

where $\lambda_{a\to b}$ is the intensity associated with the branch on the default tree that represents a movement from state a to state b, and a state of the system is defined as where a certain number of defaults have occurred in a particular order. To solve this equation, one just works along the default tree from the state 0 in which no defaults have occurred as we have demonstrated in the simple 2-firm network above.

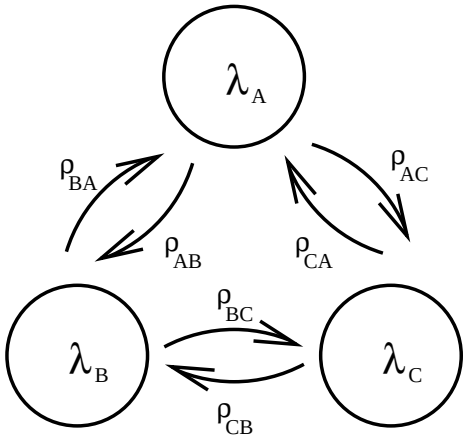


Figure 6.4: Default network for 3 reference credits A, B, and C.

6.2 Pricing a Hybrid kth to Default Basket

Now we can write down the probability of being in a particular state on a default tree at time t, we can move on to price a simple kth to default basket. In fact, because of the nature of our default-tree method we can price a hybrid basket in which payout is only made given a particular order of defaults.

Consider a simple contract which is written on three underlying reference credits A, B, and C. The contract will pay S dollars if and when credit A, *then* credit B, *then* credit C defaults before time T. It will only pay if all three assets default in that order. The general default network of the assets underlying this basket can be seen in Figure 6.4.

To find the value of the contract, all that is required is the probability of being in state AB at time t and the instantaneous probability of moving out of state AB and into state ABC. If the system is in state AB, asset C is the only surviving credit. Hence, by referring to the default network, one can see that the intensity of the new Poisson process in which only C exists, and A and B have defaulted, is just the sum of the original intensity of C, given by λ_C, and the effect upon C of A and B defaulting, given by ρ_{AC} and ρ_{BC} respectively. Therefore, the value of the contract for the general default network is just

$$V_{ABC} = \int_{t=0}^{T} (Se^{-r(t)t} P_{AB}(t)(\lambda_C + \rho_{AC} + \rho_{BC}))dt, \tag{6.22}$$

where $r(t)$ is the risk free interest rate (which here we are assuming to be determin-

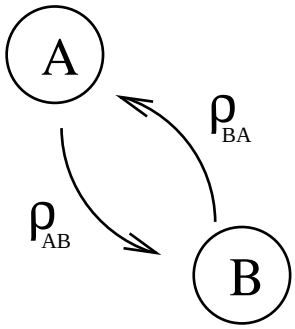


Figure 6.5: Small network of two linked firms A and B.

istic) and $P_{AB}(t)$, is given by[1]

$$\begin{aligned} P_{AB}(t) &= \frac{\lambda_A(\lambda_B+\rho_{AB})}{(\lambda_A-\rho_{AB}-\rho_{AC})}\Big[\frac{e^{(\lambda_B-\rho_{AC}-\rho_{BC}+\lambda_A)t}}{(\lambda_B-\rho_{AC}-\rho_{BC}+\lambda_A)}-\frac{e^{(-\lambda_B-\rho_{AB}+\rho_{BC})t}}{(\lambda_B+\rho_{AB}-\rho_{BC})}\Big] \\ &\quad +\Big[\frac{\lambda_A(\lambda_B+\rho_{AB})}{(\lambda_B-\rho_{AC}-\rho_{BC}+\lambda_A)(\lambda_B+\rho_{AB}-\rho_{BC})}\Big]e^{-(\lambda_C+\rho_{AC}+\rho_{BC})t}. \quad (6.23) \end{aligned}$$

Note that when we are pricing a financial instrument we use the risk neutral default intensities and not the real world default intensities. This is to ensure arbitrage free pricing. We tackle this point in more detail in Section 2.1.5.

6.3 Pricing a Simple Linked Bond

The default network shown in Figure 6.5 has the default matrix,

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda & \rho_{AB} \\ \rho_{BA} & \lambda \end{pmatrix},$$

where for the interests of simplicity $\lambda_A = \lambda_B$.

Now, let us denote by B_0^T the price of a zero coupon non-defaultable (risk-free) bond with maturity T, and B_A^T as the price of a zero coupon defaultable (risky) bond issued by firm A with maturity T. Both bonds pay a principal of 1 at maturity. Assuming a constant interest rate, r, the value of the non-defaultale bond in the reduced form intensity based pricing framework (see Section 2.1.5) is given by

$$B_0^T = e^{-rT}. \quad (6.24)$$

We also assume that upon default the bond issued by firm A will have no recovery value. Therefore, the price of the risky bond is given by

$$B_A^T = B_0^T(1-(\text{Prob } A \text{ defaults before } T)) \quad (6.25)$$

[1]The derivation of $P_{AB}(t)$ is given in appendix D.

$$= \quad B_0^T(1 - \int_{t=0}^{t=T} (P_0(t)(\lambda) + P_B(t)(\lambda + \rho_{BA}))dt), \tag{6.26}$$

I.e., the probability A defaults before time T, is the probability that we are in state 0 and we default directly into state A, plus the probability we are in state B and then we default into state A, before time T.

We can write down the probability of being in state 0 at time t as

$$P_0(t) = e^{-2\lambda t}. \tag{6.27}$$

Therefore, the probability of being in state B at time t is

$$P_0(t) = \frac{\lambda}{(\rho_{BA} - \lambda)}\Big[\lambda e^{-2\lambda t} - e^{-(\lambda+\rho_{BA})t}\Big]. \tag{6.28}$$

Hence, the value of the risky bond is just

$$B_A^T = B_0^T\Big(1 - \int_{t=0}^{t=T} \Big(\lambda e^{-2\lambda t} + \frac{\lambda(\lambda + \rho_{BA})}{(\rho_{BA} - \lambda)}\Big[\lambda e^{-2\lambda t} - e^{-(\lambda+\rho_{BA})t}\Big]\Big)dt\Big) \tag{6.29}$$

$$= B_0^T\Big(1 - \frac{1}{(\rho_{BA} - \lambda)}\Big[\rho_{BA}(1 - e^{-2\lambda T}) - \lambda(1 - e^{-(\lambda+\rho_{BA})T})\Big]\Big). \tag{6.30}$$

Above we have shown how the tree based method can provide a simple solution to the looping default problem of Jarrow and Yu which we detailed earlier in Section 3.2.

Figure 6.6 shows the dependence of the risky bond value B_A^T on the default dependence ρ_{BA}. As one would expect, the greater the dependence, the higher the value of the bond. Also note the variation of bond price with change in the default dependence ρ_{BA} here is quite small, with the dependence of price on variations in ρ_{BA} being most significant for small values of ρ_{BA}.

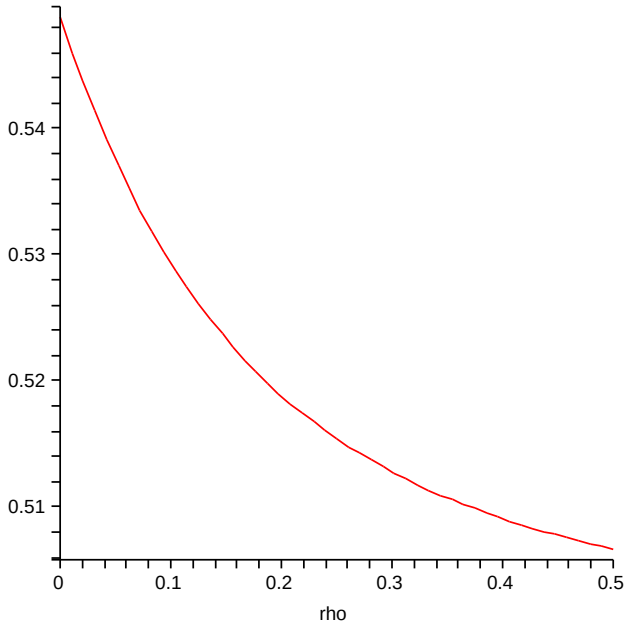


Figure 6.6: Plot of bond price versus default dependence ρ_{BA} for $r = 0.05$, $T = 10$, and $\lambda = 0.01$.

Chapter 7

Conclusions and Discussion

We have described a very simple intensity-based default mechanism in which the default dependence between firms is modelled as a directed network. Using the properties of a generalised Poisson processes, such as the superposition of independent Poisson processes, we have developed a mathematical framework for calculating the expected time for a certain number of default events to occur. We worked through from the most simple case of a network in which all vertices are unconnected, to a network in which all vertices were connected to all other vertices, until finally we considered the case of a general network. For the general network we developed a default tree in which all the possible orders of the default events were represented together with the probability of a sequence of events occurring and the associated intensities with which each of those events would take place. The idea of the default tree was taken further with a binary representation of the possible default states that can occur and the construction of a simple matrix based algorithm for calculating the expected times to be in each of the binary states. This general algorithm could then be easy implemented computationally.

Once we had a method to calculate the expected time for any number of defaults to occur on any general network, we analysed the impact of the default network structure on the expected times of default. We began by considering the most simple 3, 4, and 5 vertex networks. By comparing similar networks we tried to develop a method to predict the relative resilience to default events of different structures. This included an analysis of the in-degree and out-degree of each of a network's vertices. We found that while some predictive power into the stability of a network could be made using these simple approaches for 3 and maybe 4 vertex networks, for more complex networks a statistical analysis and description is more appropriate.

In fact, studying a range of statistical measures of much more complex network structures, including number of cycles and edge complexity, we found that something

as simple as the average path length between nodes is a very good predictor of the stability of a network to default events for our default mechanism. We then briefly considered the application of our default mechanism to β-graphs and further assessed the links between the complexity of the network structure and its response to defaults.

Finally, we looked at how to price simple baskets of reference credits and simple structures of linked bonds using the default tree methodology. Once we had shown that it was possible to calculate the probability of being in a particular state on the default tree for a given network, we then showed it is trivial to price simple credit default instruments whose default structure is given by that network. Therefore, in doing this, we have given a very simple method for the solution of Jarrow and Yu's problem of looping defaults.

We believe that our study has successfully brought together a simple intensity based default mechanism and ideas from graph theory and complex network analysis to provide a deeper understanding of what determines the stability of a portfolio of credits and its response to default events. Whether the model will prove useful as a pricing tool is more difficult to ascertain as this would rely on the accurate calibration of the model, which we have not broached in this piece of work. However, we certainly think that the application of network analysis will be invaluable in we are to truly understand how portfolios of credits behave. Our default tree approach to pricing and calculating expected default times is intuitive and simple to implement, but due to the nature of its construction it is clearly not appropriate for very large portfolios.

Natural extensions to the study would include a more sophisticated default mechanism, perhaps including some sort of time dependence in the effect of one firm defaulting upon another. Also, some form of dynamic nature to network, rather than having the default dependence fixed through the lifetime of the instrument being priced. And of course, one could consider stochastic individual default intensities, although any modifications should not detract from the default contagion mechanism of the model. As mentioned above, it would be useful to study how one might go about calibrating such a model, and what necessary simplifications would need to be made to do so, given the relatively limited amount of credit and default data in the marketplace.

Finally, applications of the model outside of finance might be modelling the spread of disease in a population or measuring the resilience of a computer network to attacks or computers crashing.

Part II

An Agent Based Network Model of Firm Default

7.1 Overview

In this part of the thesis we study a novel agent-based approach to modeling the default dependence between firms. Firstly, we review some of the work that has already taken place in this and related fields and look at some models that lend ideas to our approach. We consider what criteria need at be satisfied by any general model of firm-firm dependence and what we set out to accomplish with our attempt.

We then detail how the model is constructed, initially beginning with a simplified approach and then moving on to a more complex and hopefully more realistic version. We then make a broad analysis of the final model, comparing our results with empirical studies into real-world behaviour.

Finally we review the success of our model and consider possible applications to other areas.

Chapter 8

Introduction to the Agent Based Network Model of Firm Default

8.1 Why an Agent Model of Credit Default?

Agent based models have long be used to study the behaviour of a wide variety of complex systems. Simulating the fundamental microscopic interactions between the constituent agents can provide a deeper understanding into a system's observed macroscopic properties. The success of these models depends upon their ability to capture real-world dynamics. This must be coupled, however, with their retaining sufficient simplicity to provide a transparent view of the underlying mechanics of the system, and allow an understanding into which processes lead to which effects.

Defaults of companies have been prevalent in the news in recent years with defaults of major firms such as Enron, WorldCom, and most recently Parmalat having dramatic effects on the rest of the financial market. The possible reasons behind a default are countless - corrupt managers, terrorist attacks, recession, to name a few. However, we believe a major contribution to default risk for a firm comes from its dependence on the set of business partners upon which it is reliant to exchange goods and conduct transactions. This real-world network of business links leads to the existence of a "dependence network" connecting all firms. Crucially, the dependence network will be dynamic, reflecting changes to the network of trading partners as firms change their business associations through time. By studying the nature of the real-world business networks, and the influence of these networks on the default dependence between firms, we may arrive at a better understanding of the reasons for default and the patterns of default in credit markets.

Therefore, it seems natural to try to model the network of interacting firms as a system of interacting agents. By supplying simple rules to the agents it may be

possible to reproduce some of the collective effects seen in credit markets whilst also realistically modeling the individual dependence of a firm on its neighbours.

8.2 Producing a Realistic Model of Defaults

If one were to compose a list of properties that an agent-based model of firm default should successfully reproduce, top of that list would be a realistic picture of default dependence between the companies.

A recurrent theme in this study, and the thesis as a whole, is the idea of *asymmetrical default dependence*. Given two firms, A and B, if A were to default, there is no reason to expect the effect of this default on B to be the same as the effect on A if B were to default. There is no evidence to suggest that there exists an equal dependence between any two firms. Furthermore, given the default of a firm, whilst some companies would be negatively affected by this, resulting in them having an increased probability of default (*negative default dependence*), it should also be possible that some firms benefit from the default. These firms would experience a reduction in their own probability of default and therefore imply a *positive default dependence* between the firms. As an example of a situation where this might occur consider a small oligopoly of, say, three chocolate bar manufacturers. If one were to default, perhaps because of a problem in another area of its business such as fizzy drinks, then that would leave a greater market share for the two remaining competitors, hence a greater income from sales and therefore a reduced probability of default. A realistic model of default should be able to reproduce negative, positive and asymmetrical default dependence.

A successful model taking into consideration the above types of default dependence should hopefully lead to an accurate recreation of some collective default effects. One such effect is *default contagion* (which we refer to in Section 2.1.6). This is the action of a default in a market inducing further defaults, which then propagate still more defaults, i.e., the same process as a contagious disease passing through a population. Also, there is the idea of *default clustering*. If one looks at the pattern of a given set of default events through time, it is believed that there is clustering within this distribution. There are a variety of reasons for this, including cyclical market conditions, but another reason may lie in the default dependence structure of the firms in the market. If the model of the default dependence between the agents is realistic these other default "stylised facts" should appear as a result of this.

8.3 Current Research

To our knowledge there do not exist any dynamic agent based network models of credit default. However, under this "umbrella" topic there is much research into its constituent areas which include; dynamic networks, firms modelled as agents, models of extinction, and credit.

A seminal paper by Barabasi and Albert looked at modelling the power-law degree distributions of systems such as the World Wide Web and genetic networks [34] [1]. Their dynamic networks were constructed based upon two mechanisms; networks expand continuously by adding new vertices, and, new vertices attach to sites that are already well connected. Based on these two simple rules they formulated an algorithm to construct dynamic networks which successfully reproduced a power-law distribution of vertex connectivity.

Another field with extensive study into dynamic networks is the modelling of social networks - networks of acquaintance between people. Jin, Girvan, and Newman developed a series of growth algorithms which model how social networks grow and how their resulting structure develops [35]. Even though their algorithms are very simple, their networks demonstrated the desired effects, such as high levels of clustering and community structure, but, unlike the networks of Barabasi and Albert, no evidence of power-law degree distributions (which they believe not to be present in social networks).

Robert Axtell has done a large amount of work on the application of agent based simulations to the modelling of firm dynamics. In [36], Axtell models the evolution of the firms themselves, each of which is made up of a system of interacting agents. These agents can migrate between firms and even start up new firms as the economy evolves. He studies the growth rate of these firms, their lifetime distributions and looks for evidence of a power-law relationship in the distribution of firm sizes.

Perhaps more relevant to our study, a paper by Thurner, Hanel, and Pilchler, proposed an iterative risk-trading game between banks modelled as a population of heterogeneous agents [37]. The financial system in which the agents trade is modelled using different network topologies, such as 1D and 2D lattices, random networks, and fully connected networks. Alluding to the Basel (II) Accord[2] they then measured the dependence of different regulatory actions on the different network topologies. A key difference between this model and our own is that the connections between the agents

[1] A degree distribution is a histogram of the number of neighbours to which each vertex is attached.

[2] The Basel (II) Capital Accord is a framework that aims to regulate the amount of credit risk financial institutions can take onto their books and how they should cushion this risk against default.

are characterised by a graph topology that is specified exogenously, and, hence, not a result of the interactions between the agents themselves. Also, the graph structure remains constant over time with the only change to the network being the default of the banks sitting at each of the vertices. The agents in this model are purely selfish and do not learn, nor have any incentive to cooperate with each other.

The model is made up of N sites on a defined network, and upon each site sits a bank that receives trading requests from an external client. Banks must then trade with the other banks to which they are connected to reduce their risk exposure down to some desired level. The banks are heterogeneous as each is given a randomly selected risk aversion parameter. As banks move risk between themselves, regulatory parameters may mean some banks are unable to trade away a sufficient amount of risk and this may lead to a propagation of illiquidity within the system. Each bank has some wealth parameter based upon the size and success of the trades it makes with neighbouring banks and their own external clients. If this wealth falls below zero the bank is said to have defaulted and can no longer trade as before with its neighbours. Hence, the default of a bank may quickly force neighbours to default as well, producing contagious defaults as mentioned above. Amongst their findings they showed, perhaps expectedly, that the better connected a network the lower the volatility in the agents' wealth and also that a tightening of regulation on the banks does not necessarily result in an increase in the safety of the banking system as a whole.

Briefly moving away from the topic of finance, Mark Newman proposed in [38] a model for the extinction of species. His model is based around the idea of coevolution which arises as a result of the interactions between different species - the adaption of one species forcing the adaption of others, for example, the evolutionary pressure the cheetah and the antelope place on each other to run faster. The model postulates that many species coevolving together can cause mass extinction of certain species whose livelihoods are destroyed by the changes to the system. Also, pseudo-extinction (which is the replacement of a species by its own descendants) may arise because of coevolution between species. It is difficult not to draw the obvious parallel here between the extinction of a species and the default of a firm, liken coevolution to some driver of default dependence, and compare pseudo-extinction with the competition from new companies on established firms.

In the Newman model, there are N agents, each with a stress tolerance x_i, chosen at random from a uniform distribution. At each iteration an external stress on the system given by η is selected from a stress distribution, $P_{stress}(\eta)$. All those agents

where $x_i < \eta$ become extinct and are replaced by new agents of which a small fraction f have a new x_i. The number of species that become extinct at each iteration is then studied for different distributions of external stress.

Indeed, the parallels between the Newman model and credit default did not go unnoticed and in [39], Smith, Ormerod and Johns propose a model of firm default based around the above model of species extinction. Their model is almost identical to that of Newman. However, in addition the agents, which in their model represent firms, are placed on different types of network. If one firm defaults then there is a probability that its neighbours will default as well. The types of network studied included small world, random, and torus. Again, note that these networks are imposed upon the agents and are constant throughout the simulation. As with Newman, they looked at histograms of the size of the groups defaulting. They found that no power-law dependence was present in the distributions of defaults per iteration, and also, perhaps surprisingly, that the model showed little sensitivity to the network topology connecting the agents.

8.4 Initial Considerations on Model Design

At the outset of the model design we specified a list of criteria which the model must include. These were:

- Dynamic network - agents must be able to change neighbours;
- Represent firms as heterogenous agents;
- Agents need some way to trade with each other;
- Agents must possess a measure of performance;
- Agents must be able to default given bad performance;
- A default-dependence must exist between agents.

We decided that if a firm is represented by an agent, then each agent must have some individual set of trading requirements that it would look to satisfy to maximize some internal measure of score or wealth. Then, as with the model of Thurner *et* al, if this wealth moves below some level then that agent or firm is deemed to have defaulted [37]. Agents must therefore be able to select which of the other agents they trade with and, so that the network is dynamic, agents must continually review the value of each of their links and have some method for assessing the potential of

forming new trading relationships. This would allow the agents and the network to adapt to defaults.

A model by Martino and Marsili modelled firms as heterogeneous agents, where each firm was characterised by a strategy detailing in with of C commodities it was a producer or consumer [40]. This idea naturally lent itself to our model as agents could then move though the population assessing the strategies of other agents and linking to those which maximised some reward based upon the requirements of their own strategy. This would also mean that if an agent were to lose a neighbour due to default, its own rewards might be affected as it searched for possible new links - implying a default dependence between the agents.

Finally, given the work of Axtell on distribution of firm lifetimes, by including some method to introduce new agents as others default, it would be possible to study the evolution of a whole economy of firms through time.

Chapter 9

Model Overview

In the Agent Based Network Model of firm defaults firms are represented as a population of N agents.[1] Each agent holds one strategy which represents technologies in which the agent is a consumer, a producer, or neither. Agents must find partners with whom to trade so as to maximize their potential payoff at each iteration of the simulation. Each agent's performance must be above some threshold, otherwise they are considered to have defaulted and can no longer trade or participate in the simulation. An agent's trading partners can be considered neighbours on a network with agents or firms represented as nodes and a trading relationship between firms as a link.

Each iteration of the simulation can be broken down into three main stages. Firstly, agents trade with their neighbours, and depending partly upon their choice of neighbours their scores are incremented appropriately. Secondly, after each agent has had a chance to trade, agents are checked to see if they satisfy the criteria that constitute a default. Finally, agents are offered new potential trading partners and given the chance to assess and adjust their current links to other agents. This procedure is then repeated many times, beginning again with agents trading with each other, see Figure 9.1.

9.1 Agents and Strategy Vectors

Each agent i $(i = 1, ..., N)$ holds one strategy vector R_i. A strategy vector consists of q technologies, where each technology can take a value $+1$, -1, or 0. This value indicates whether the agent is a producer or consumer of each technology, see Table 9.1 and Figure 9.2. As mentioned in section 8.4, the idea to characterise a firm with

[1] We shall also refer to this model as the agent network model.

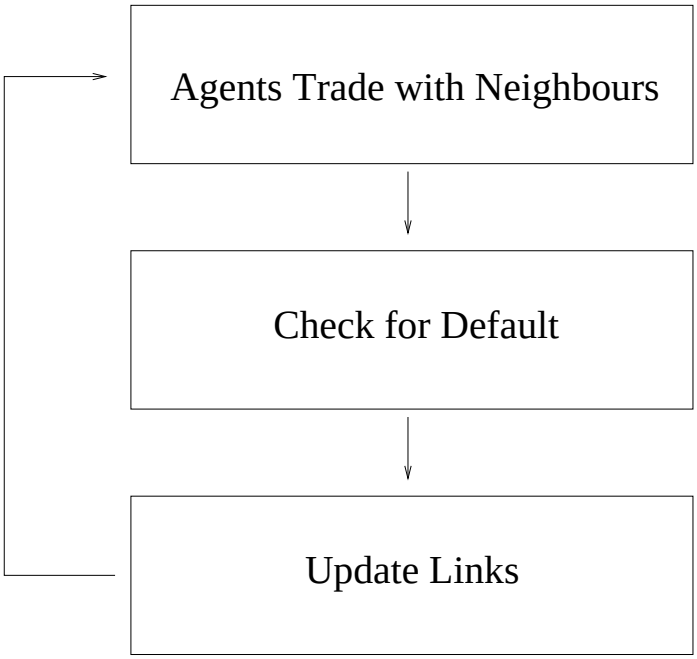


Figure 9.1: Flow diagram detailing the three main stages of the agent based network model of firm default.

a vector of technologies is not new and has been used before in the study of large random economies [40].

+1	Producer of technology
-1	Consumer of technology
0	No trade - neither consumer nor producer

Table 9.1: The three different possible values a technology can take.

As an individual firm is represented as an agent, the strategy vector can be thought of as a description of that firm. For example, if every firm in the simulation were in the automotive industry, then each technology might refer to a different component or process in car manufacture, see Figure 9.3. In this simple example one can see that the major car manufacturer is a consumer of tyres and electronics whilst also a producer of engines and complete cars. The specialist sports car manufacturer is similar to the major car manufacturer although they do not have the expertise to build their own engines and hence they are also a consumer of these. The individual consumer - *the man on the street* - is a consumer of complete cars only.

In the simulation, each agent's strategy is created by randomly generating a -1 or $+1$ value, with equal probability of either occurring, for each of the q technologies.

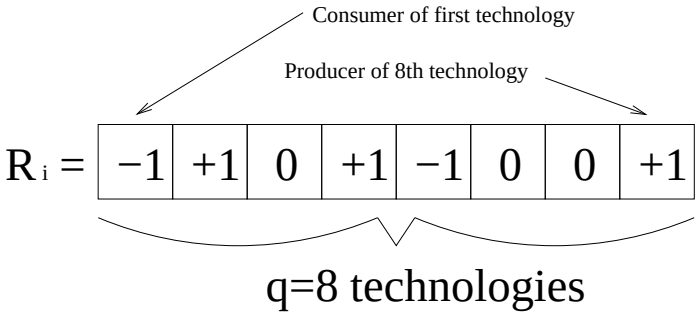


Figure 9.2: An example of a typical strategy vector.

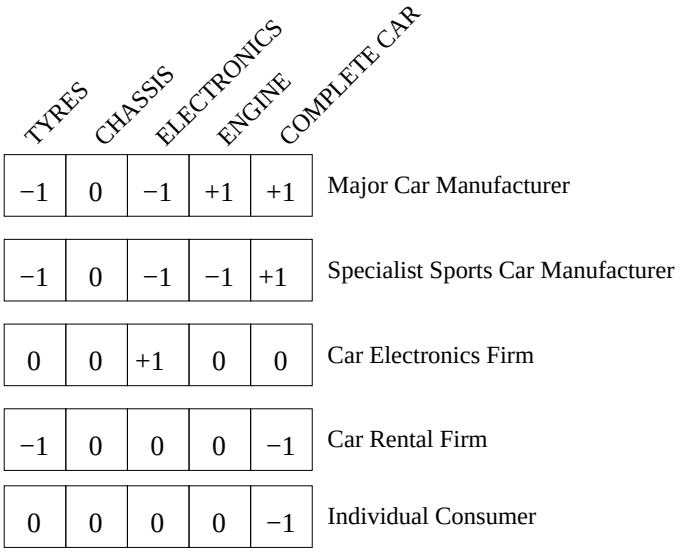


Figure 9.3: Very simplified representation of car industry to illustrate idea of technologies in a strategy.

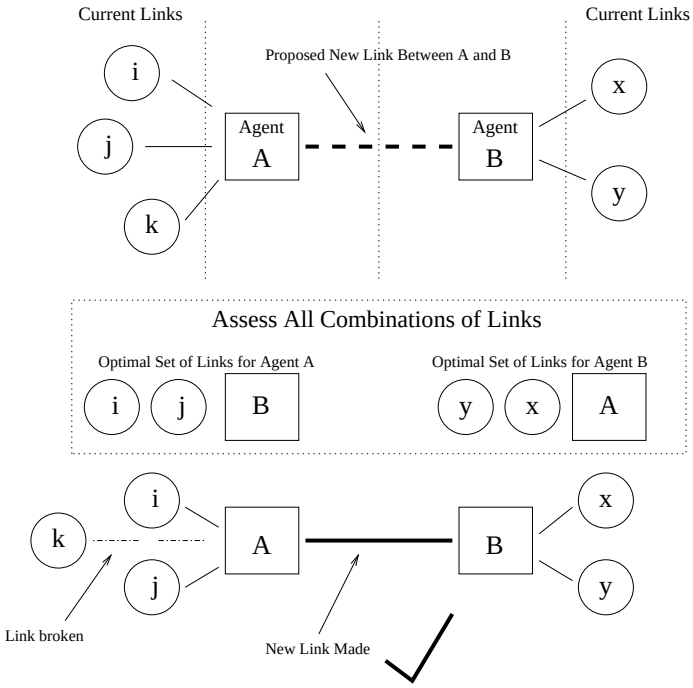


Figure 9.4: Agents A and B both decide whether a new link should be constructed between them. As each agent has the other in their optimal set of neighbours this link is made.

Then, each technology value is set to zero given some probability z, which is the same for all agents.

9.2 Network Formation and Link Adjustment

At every iteration of the simulation each agent is assigned a possible new link with one other agent. These links are randomly selected from a uniform distribution over all active (as opposed to defaulted) agents. The agent will then assess whether it wishes to make the link, as does the new proposed neighbour. Both agents review all their current links along with the new proposed link and select the combination of neighbours that will maximize their potential payoff. If both agents select each other as part of their optimal set of neighbours then the link is made, see Figure 9.4. Note, a link is only made if *both* agents decide it is optimal. Also, an agent will only make adjustments to its current set of neighbours if it is also to make a new link to a new neighbour.

Initially, agents are linked only to themselves and at the first round of trading will play against themselves only. However, over time a network of linked agents will

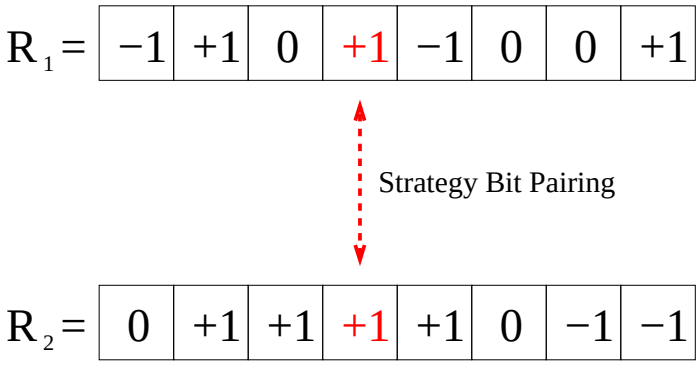


Figure 9.5: Illustration of a strategy bit pairing.

evolve that can be represented as an undirected graph described by the adjacency matrix $\underline{\underline{A_t}}$.

9.3 Agent Scoring Mechanism

Each agent i has a score $s_i(t)$ which can be thought of as modelling the firm's wealth. In the first part of each iteration of the simulation, when agents trade with their neighbours, agents are rewarded by comparing their needs in each technology with those of each of the agents they are linked to. Looking at every active agent in turn, for every link that each agent has, each strategy bit pairing (see Figure 9.5) is compared and the agent's score, $s_i(t)$, is incremented appropriately. The scoring for each bit pairing is given in Table 9.2.

Bit Pair Combination	Score
$+-$ or $-+$	$+2$
$++$ or $--$	-2
00	0
$+0$ or $0+$ or $0-$ or -0	-1

Table 9.2: Scores for different combinations of strategy bit pairings.

In the example below, agent a is linked to agent b and agent c, see Table 9.3.

Firstly agent a will compare each of its strategy bits with the appropriate bit in agent b, and then again with agent c, see Table 9.4, using the scoring weights given in Table 9.2.

By summing the rows of Table 9.4, one can see that agent b contributes a total of 2 to the score of agent a while agent c contributes 0.

Agent	Agent Strategies			
a	$+1$	-1	0	$+1$
b	-1	-1	0	-1
c	$+1$	0	$+1$	-1

Table 9.3: Strategies for agents a, b and c.

Agent Pairing	Score Contribution for Bit Pairing			
A to B	$+2$	-2	0	$+2$
A to B	-2	0	0	$+2$

Table 9.4: Score contribution for each strategy bit pairing.

The scoring weights given in Table 9.2 were chosen to reflect to some extent the value a firm might place on being linked to another firm. Remember that a positive strategy bit means the agent is a producer of a particular technology and conversely a negative value indicates a consumer of a technology. Therefore, if a firm is a producer (consumer) of a technology and links to a firm that is a consumer (producer) of that technology, then this is ideal for both agents and each is rewarded with $+2$ points when it is their turn to assess the score contribution of their links in the trading phase of the simulation.

However, if a firm is a producer of a technology and links to another firm that is also a producer of that same technology then this is bad for both firms as neither firm will be able to trade this technology with the other as neither has a requirement for it. Not only is the firm linked to someone to whom they cannot pass on their product, but also the other firm produces the product themselves and also wants to pass it on. So in effect, the link may be helping to support a firm who for that technology is a direct competitor. This situation is therefore given a score contribution of -2, as is the case for the pairing of two consumer strategy bits.

If a firm that does not produce or consume a technology links to a firm that is either a producer or consumer (or vice versa) then this is still not an optimal use of a link between the firms but is considered not as bad as the above situation as they are not direct competitors in that technology. This pairing is given a score contribution -1. For the case of neither firm being a producer nor consumer of a particular technology then this is scored as 0.

Note that there is a symmetry in the above scoring mechanism; a firm will be penalised if it is not a producer or consumer of a particular technology and it links to an firm that is, *and*, a firm is penalised equally if it is a producer (or consumer) and

it links to a firm that is neither. This means that a firm that may only produce or consume one technology, such as the car electronics firm in Figure 9.3, will be heavily penalised unless it can find someone who also only has a need to buy (or sell), in very few technologies, i.e., has a large number of zero strategy bits.

To introduce a stochastic component to the score evolution of each of the agents, each strategy bit pairing is only considered given some probability β which is held constant throughout the simulation and is the same for every agent. For $\beta < 1$, if an agent does not change its links for some time its score does not just increase linearly at some constant rate but rather is subject to random fluctuations. This randomness can be thought to model a host of reasons for particular trades between firms not taking place for certain technologies, other than the firms ceasing to be linked altogether. Such reasons may be lack of raw materials or striking work forces, etc.

For the neighbour assessment phase of the simulation, agents assess every possible combination of their neighbours and the new proposed link using the above scoring mechanism. It is clear in this simple case, given an agent wants to maximize the potential score it can gain at the trading phase of the simulation, that it will always choose to link to a new agent if the sum of the score contributions for each strategy bit pairing is greater than zero. If the sum of the score contribution is zero then it does not care whether it will link or not. If an agent has more than one combination of potential neighbours that gives the same maximum potential payoff then it will choose randomly which combination to use as its optimal set.

9.4 Modeling Firm Default

As shown in Figure 9.1, after the agents have traded with each other, each is looked at in turn to see if it satisfies the criteria for default. For an agent to default its score $s_i(t)$ must move below some value indicated by the default level $D(t)$:

$$D(t+1) = D(t) + \alpha \tag{9.1}$$

$$D(0) = \delta, \tag{9.2}$$

where α is positive and constant and δ is the initial default level for all agents. Hence agent i defaults when:

$$s_i(t) < D(t). \tag{9.3}$$

Note the parallels here between this default mechanism - a firm is considered to have defaulted if its score moves below some threshold - and the structural credit models.

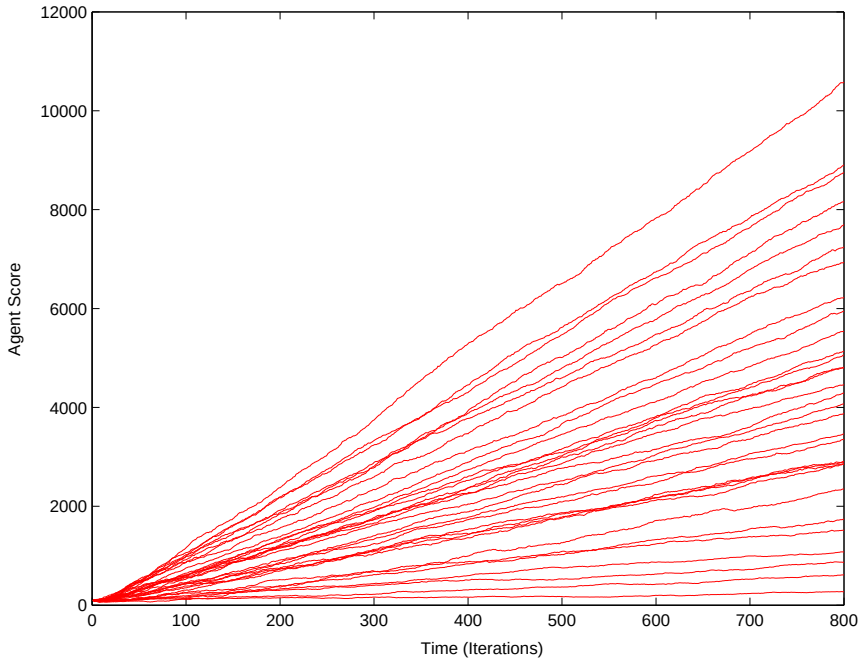


Figure 9.6: Score evolution of agents over one run of the simulation. Here the default level growth α is set to zero. $N = 31$, $q = 9$, $s_i(0) = 100$, $z = 0.2$, $\beta = 0.25$, $D(0) = -1000$, $\alpha = 0$.

When $\alpha > 0$ an agent's score $s_i(t)$ must on average grow over time at a rate greater than or equal to α at each iteration to avoid default. As soon as an agent does default it is removed from the simulation and all links to other agents are broken. Initially, δ is usually set well below the agents initial scores, $s_i(0)$, to allow the agents time to form new links. This default mechanism is referred to as Common Threshold Default or CTD.

9.5 Initial Results

Figures 9.6 and 9.7 show the evolution of agent scores over two different runs of the simulation.[2] All parameters are the same for both runs except in Figure 9.6 the default level growth rate α is set to zero while in figure 9.7 it is greater than zero.

In Figure 9.6 one can see the approximate constant growth rate of each of the agent's scores, $s_i(t)$. There is little change in the relative position each agents occupies in terms of total wealth at any one time. There is some randomness in the score

[2]In Appendix F there are tables of all symbols and terminology used in the Agent-Based Network Model of firm default.

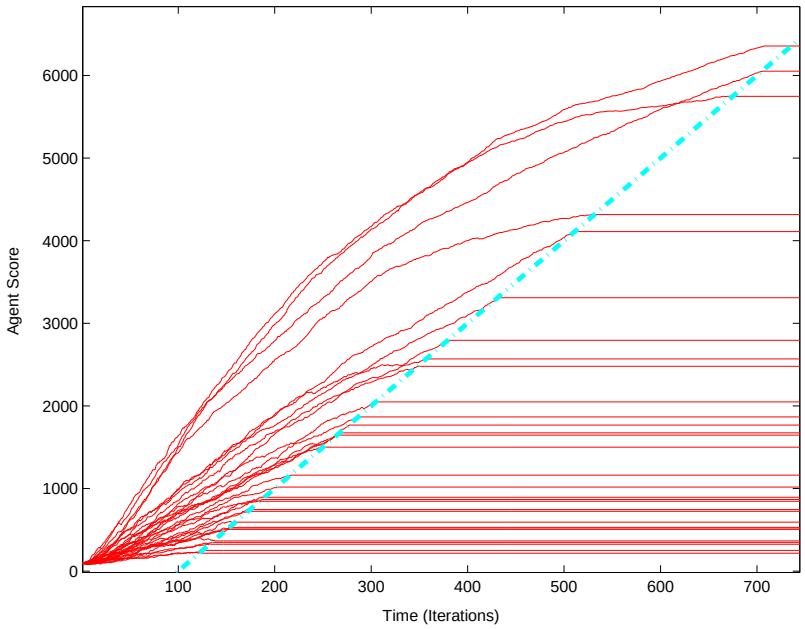


Figure 9.7: Score evolution of agents over one run of the simulation. Here the default level growth α is not set to zero. $N = 31$, $q = 9$, $s_i(0) = 100$, $z = 0.2$, $\beta = 0.25$, $D(0) = -1000$, $\alpha = 10$.

growth due to the parameter β - which is the probability a strategy bit pairing will be checked - being fairly low.

The effect of a setting $\alpha > 0$ is shown in Figure 9.6. The strong dashed line marks the rising default level. There is a significant reduction in the rate of growth of agent scores as defaults begin to occur, suggesting a default dependence between the agents. There is also more frequent changing in the relative position of each agent in terms of wealth. This is due to the default mechanism removing agents upon which others are dependent and hence causing the reordering of links between agents as agents search for new optimal neighbour combinations from the set of undefaulted agents.

Chapter 10

Towards a More Realistic Agent Based Network Model of Firm Default

Detailed below are some modifications to the simple model described above. These modifications bring the model closer to what one might consider as more realistic of real world firm interactions.

10.1 Revisiting the Strategy Vector R_i

Above, an agent's strategy, R_i, is made up of just $+1$, -1, or 0 values, indicating whether they are producers, consumers, or neither, of each technology. This does not indicate what quantities of each technology they wish to buy or sell at each iteration. In the above version of the model, for large N there is potentially no limit to the number of neighbours an agent can satisfy. Clearly this is unrealistic as firms have different sizes and hence different capacities for consumption and production. The *man on the street* only needs a very small number of cars while a car manufacturer will (hope to) sell many.

Here we introduce a modification to the agents' strategy vectors and scoring mechanisms - to incorporate the idea above of each agent having a maximum potential market share for each technology. Note that the modified scoring mechanism is used in the agent trading *and* link assessment phases of each iteration of the simulation, see Figure 9.1.

An agent's strategy vector R_i is revised so it still consists of q technologies, but each technology can take any *integer* value. Strategy bits are drawn at random from a uniform distribution between 0 and $\pm\mu_{i,max}$. Here, $\mu_{i,max}$ is the maximum value

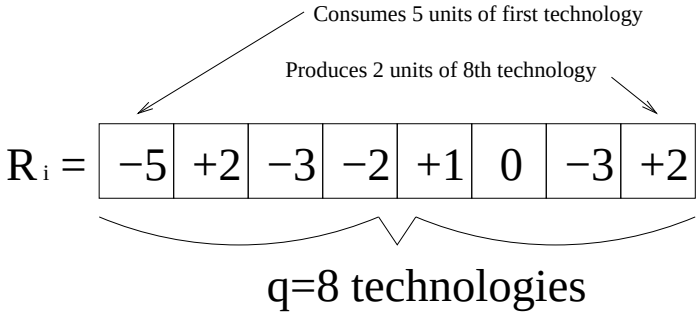


Figure 10.1: An example of a typical modified strategy vector with integer bit values.

each agent's strategy bit can take and can be thought of as indicating the size of each firm - its capacity to produce or consume. For each agent, $\mu_{i,max}$ is randomly selected from a uniform distribution between κ_{max} and κ_{min}, the firm size parameters, which are held constant over the simulation. As before, each strategy integer bit[1] is then set to zero given some probability z. A typical modified strategy vector R_i is shown in Figure 10.1.

10.2 Asymmetrical Linking

Another modification to the earlier model is that an agent is no longer penalised for appearing to over-fulfill its needs. By this we mean that, as stated in Section 9.3, if an agent is neither a consumer or producer in a particular technology, it is still penalised if it links to an agent that is active in that technology. Now, if an agent is not a producer or consumer in a technology, *or* has already allocated all of it capacity to trade in a particular technology to other agents in its neighbourhood, it is *not* penalised for making further links which it cannot satisfy. This introduces an asymmetry into the scoring mechanism - an agent *is* penalised if it cannot satisfy all if its needs, although it *is not* penalised for making a link that appears, in effect, to "over-satisfy" its requirements.

The justification for this modification is that realistically a consumer such as the *man on the street* is not penalised if he buys just one car from a manufacturer and none of its other products such as spare engines etc, whereas a car manufacturer will look to find enough people to buy all their cars and will be penalised for those it produces but does not sell.

[1]Strategy bits will now also be referred to as *integer strategy bits* to highlight, where necessary, that each strategy element can now take an integer value.

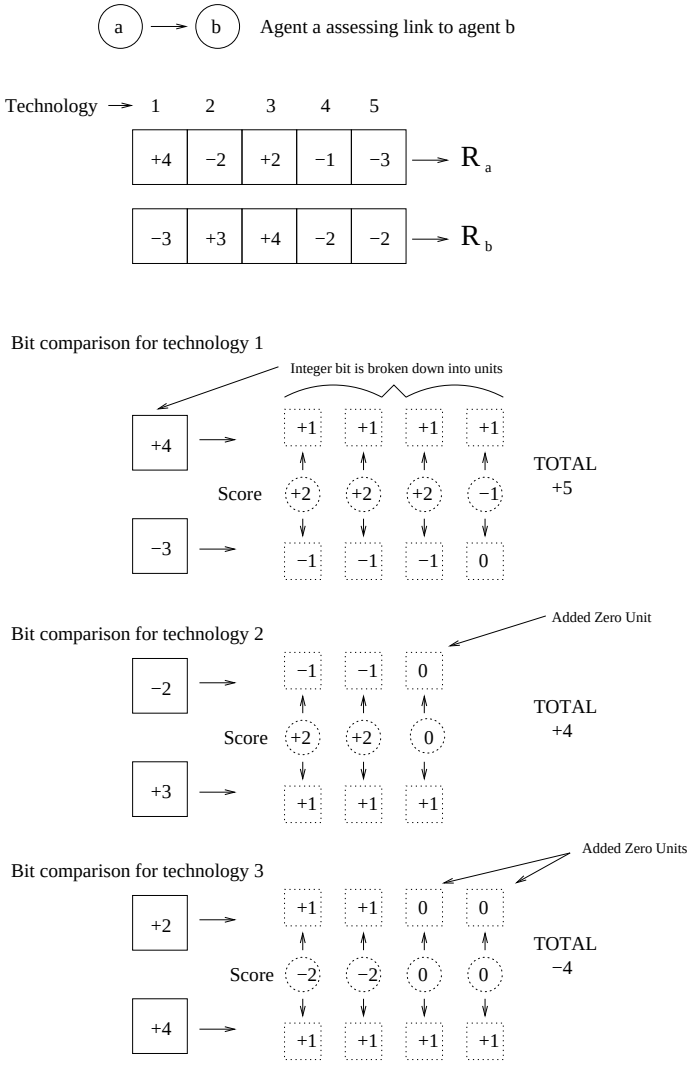


Figure 10.2: Comparison of two strategy vectors. Agent a is assessing the value of linking to agent b.

10.3 A Modified Scoring Mechanism

To incorporate the concept of market share, in the strategy bit comparison each integer bit is broken down into single $+1$, -1, and 0 units. Each unit in turn is matched against the units of the strategy being compared in a similar method to the original scoring mechanism. The modified scoring mechanism is demonstrated in Figure 10.2. Here, agent a is linked (or considering linking) to agent b *only*. Each strategy bit is compared in turn.

As one can see from Figure 10.2, the strategy bits of agent a are broken down into units and each unit is compared in turn with the corresponding units for the strategy

bit belonging to agent b. Any missing units are set to zero. The scoring weights for each possible unit pairing are given in Table 10.1. Note the last 2 rows of this table which illustrate the practical implementation of the asymmetrical linking referred to in Section 10.2, given by the different score for unit pairings of $+0$ (or -0) versus $0+$ (or $0-$).

Unit Pairing	Score Contribution for Unit Pairing
$+-$ or $-+$	$+2$
$++$ or $--$	-2
00	0
$+0$ or -0	-1
$0+$ or $0-$	0

Table 10.1: Score contribution for each possible unit pairing. First unit belongs to agent making comparison, second unit belongs to agent to which it is linked (or considering linking).

In Figure 10.2, from the strategy bit comparison for technologies 1 and 2, one can see the asymmetrical treatment of the zero unit, depending upon whether it belongs to agent a or agent b.

The relevance of the asymmetrical treatment of zero units is clearer when comparing a set of more than two linked agents, as shown in Figure 10.3. Here agent a is linked (or considering linking) to agent b *and* agent c. Unlike the simpler version of the model where each link could be assessed and scored in turn, due to the introduction of the market share idea an agent must assess a set of neighbours (whether actual or potential) as one group. As shown for technology 1, the units of agent b and agent c are combined. As with the previous example, so that the number of units in both groups are equal, any missing units are assumed to be zero and added to the *end of either row* for comparison. Hence, a zero unit is added to the end of agent a's strategy bit breakdown for technology 1.

The units for agent b and agent c can be reordered (as shown for technologies 4 and 5) so that agent a can gain the maximum potential score from a link for a particular technology. Hence for technologies 4 and 5 agent a matches its first unit to agent c to gain 2 points. Then, for technology 4, the remaining units of agent a are paired to the units of agent b. Agent a is not penalised but nor does it gain from not being able to supply units to agent b. For technology 5, agent a is penalised for matching its remaining negative units to the negative units of agent b.

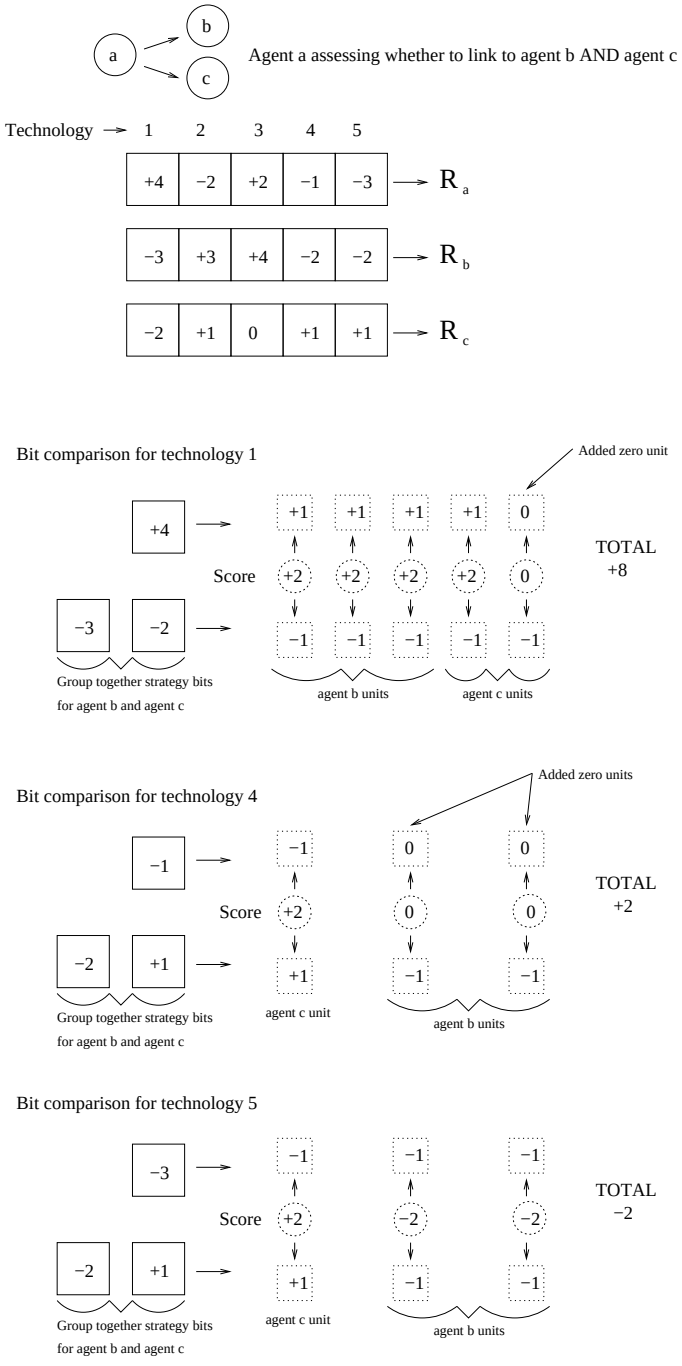


Figure 10.3: Comparison of three strategy vectors. Agent a is assessing the value of linking to both agent b and agent c.

This ability to reorder neighbours when assessing scores is in line with what one might expect in the real world where a firm will (or should) always try to maximise their payoff from a set of links to other firms.

Again, as with the simpler model detailed before, in the first part of each iteration of the simulation - when agents trade with their neighbours and their scores are incremented - each agent's strategy bit is assessed with a probability β. However, in the third part of each iteration - when agents assess all possible combinations of their current neighbours and the new proposed link - all strategy pairs are compared.

10.4 An Alternative Definition of Agent Default

Referring back to the initial results for the simpler model of Chapter 9 and Figure 9.7, one can see that for an agent to default its score must cross some threshold common to all agents. This may be unrealistic, as even if a firm has been performing very well historically, the loss of some significant trading links to competitors over a few consecutive quarters may be enough for the company to default.

Therefore, rather than use a default threshold, an alternative default mechanism is to look at the change in growth of each agent's score over time:

$$s_i(t) - s_i(t-g) \geq 0, \tag{10.1}$$

where $s_i(t)$ is the score for agent i at time t, and g is the time period over which the change in agent score is measured. Agents are given a certain number of iterations before the change in the growth of their score is assessed, which is given by the parameter g_{start}.

Setting the score growth level at zero is arbitrary and the parameter of interest is over how long the score growth is assessed, given by g. If g is large, then the agents are far more resilient to periods of negative score growth. If g is small, only a few consecutive iterations of negative growth may result in default.

This default condition is called Score Growth Default or SGD.

10.5 Cost of Existing

As one might expect, a large firm will in general have larger operating costs (staff wages, rent, servicing debt etc.) than a smaller firm. Because of the introduction of the market share idea and the parameter $\mu_{i,max}$, which effectively governs the size of each firm, it seems fair to add a cost of existing, ϵ, to the model. By changing the

value of ϵ one may penalise a firm to a lesser or greater degree depending upon how large it is.

The cost of existing results in each agent's score being incremented at each iteration by some amount given by:

$$-\epsilon\beta \sum_{k=1}^{q} |R_i(k)|, \tag{10.2}$$

where ϵ is the cost of existing, β is the probability that each strategy bit is assessed, q is the number of technologies or strategy bits, and R_i is the strategy score vector for agent i. By including β, the cost of existing is scaled to the amount by which the agent's score is incremented in the trading phase of each iteration.

Given the nature of the scoring mechanism employed, the maximum potential score for an agent is directly related to $\sum_{k=1}^{q} |R_i(k)|$. The existing cost, ϵ, is therefore used as a control parameter upon the rate of agent score growth. This is useful when using the CTD default mechanism as it allows agents with large scoring potential to still have a realistic chance of default (relative to that of the agents with lower scoring potential) if they temporarily loose some significant links to neighbours.

10.6 Cost of Linking

One may expect that firms in the real world have to pay some cost to link to another firm. Whether it is the physical cost of a leased phone line for trading or even the corporate entertainment costs of keeping a business relationship alive, it does seem plausible that firms pay some fee for each of the links they have. Also, it may be reasonable and realistic to expect a firm to choose business partners that can service as many of their needs as possible in one link - use one large firm rather than go to lots of smaller firms to do the same job.

Therefore, the simulation also includes a cost of linking, λ. The cost of linking is a cost to each agent, i, at each iteration for each of the links it has at that time, and is given by:

$$-\lambda\beta \sum_{j=1}^{N} \underline{\underline{A_t(i,j)}}, \tag{10.3}$$

where λ is the cost of linking, β is the probability that each strategy bit is assessed, N is the total number of agents, and $\underline{\underline{A_t}}$ is the adjacency matrix for the undirected graph of agent links at time t. Again, the cost of linking is scaled to the probability that a strategy bit is assessed in the trading phase of each iteration.

Unlike the cost of existing, ϵ, the cost of linking is calculated for each possible combination of current and potential neighbours when an agent is assessing the validity of making a new link. This means that an agent will not make any links to agents that do not, on average, at least pay for the cost of the link itself. For a high link cost an agent is strongly discouraged from making too many links to other agents that only fulfill a small part of its needs and will look to the so called larger agents as *one-stop-shops*.

Chapter 11

The Constant Active Agent Model

The Constant Active Agent Model (CAAM) is a version of the modified model detailed in the previous chapter where the number of active (not defaulted) agents at any one time is held constant. Therefore, whenever an agent defaults a new agent is created to take its place. This gives scope to study the impact of a variety of possible new agent types on the evolution of the system over time.

11.1 New Agent Types

In the CAAM, newly created agents can be generated at random in the same way that agents are when the simulation is initialised, *or* they can be based on the agent that has just defaulted which they are replacing, *or* they can be based on one of the best performing agents. The ratio which determine of which type the new agent is, is given by

$$r_{rand} : r_{dead} : r_{best}, \tag{11.1}$$

where r_{rand} refers to a randomly generated agent, r_{dead} to an agent based on the defaulted agent, and r_{best} to an agent based on one of the best performing agents. Best performing refers to the highest agent score $s_i(t)$ at that time.

For the new agents based upon another agent, $\mu_{redrawn}$ is the probability that $\mu_{i,max}$, the maximum possible absolute value each strategy bit can take, is redrawn at random from a uniform distribution between κ_{max} and κ_{min}, or is kept the same as that of the agent upon which it is based.

New agents based on the agent that has just defaulted have a probability q_{dead} that each bit will be redrawn at random as in the game initialisation. New agents based on one of the best performing agents have a probability q_{best} that each bit will be redrawn at random. The size of the group of best performing agents, from which

one is selected randomly from a uniform distribution, is given by g_{size}. For example, if $g_{size} = 3$, then the agent upon which the new agent is based is randomly drawn from the top *three* performing agents.

For the SGD mechanism, the initial score given to the new agents is the same as at the beginning of the simulation, i.e., equal to $s_i(0)$. For the CTD mechanism the initial score is set $s_i(0)$ above the current default level $D(t)$.

11.2 Illustration of the CAAM

In figure 11.1 one can see the score evolution of 21 agents for the CAAM. The default method used here is the score growth default or SGD. Highlighted in bold is the default and re-creation of a few generations of the same agent. When the score growth becomes negative the agent score is set to zero (note the nearly vertical lines), indicating a default. At this point a new agent is created filling the place of the defaulted agent. Again, when this agent defaults another new agent is created and so on *ad infinitum*. Hence, there a constant number of active agents at any one time until the end of the simulation. Also, note that due to the initial grace period at the beginning of the simulation and upon each new agent creation, only after which agent performance is assessed (given by $g_{start} = 20$ and $g = 10$ respectively), it is possible for agents to get negative scores before defaulting or recovering and showing positive growth.

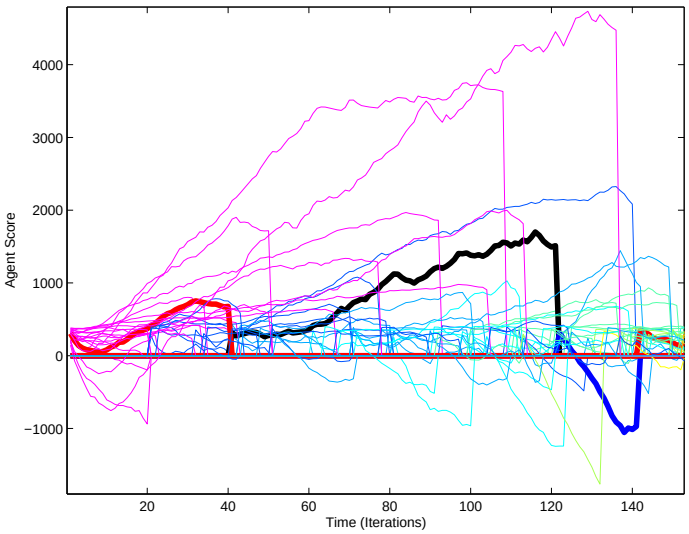


Figure 11.1: Score evolution of agents over one run of the simulation. Bold line highlights the default and re-creation of a few generations of the same agent. $N = 21$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 20$, $g = 10$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 3$, $\kappa_{max} = 20$, $\mu_{redrawn} = 1.0$, and $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$. 1st generation agents are magenta, then subsequent regenerated agents are blue through to cyan, green, yellow, orange, and finally red. For highlighted agents, 1st generation is red, 2nd generation is black, and 3rd generation is blue.

Chapter 12

Results and Analysis

Here we present a detailed study into the properties of the agent based network model of default. Looking at both the regular modified and CAAM versions of the model we gain an understanding of the complex behaviour of the evolving networks and some insight into the driving factors behind the models. We also try to marry the results to what one might expect to observe in the real world, and appraise the validity of these simulations as models of real firm-firm dependence and interactions.

While this analysis is far from exhaustive, it does provide a broad look at many of the properties of these models and should give the reader a good understanding of, and some intuition into, the evolution of these complex dynamical systems.

12.1 Comparison of Default Mechanisms

Figures 12.1 and 12.2 show the evolution of agent scores for the CTD and SGD default mechanisms respectively. At first glance both appear to produce similar results. However, if one considers the bold red and black lines in both graphs (where red highlights an agent that defaults, and black an agent that survives until the end of the simulation) one can see that for the SGD mechanism, the defaulting agent (red line) is actually the best performing (in terms of agent score) before it defaults. This agent would have been one of the last to default under the CTD mechanism as one can see by comparing the positions of the red and black lines in Figure 12.1 . Also, if one considers the bold black line in Figure 12.2, this agent outlives others which have higher agent scores before they default.

The reason for the contrasting behaviour shown in the two figures is because of the different actions of the default mechanisms as detailed earlier - one based on an overall default level, the other on individual score growth. However, whether one is

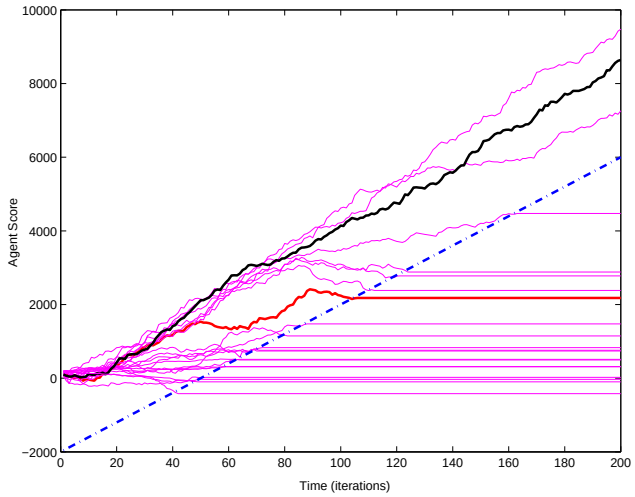


Figure 12.1: Score evolution of agents over one run of the simulation with $N = 21$, $q = 9$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.25$, default type is CTD, $D(0) = -2000$, $\alpha = 40$, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

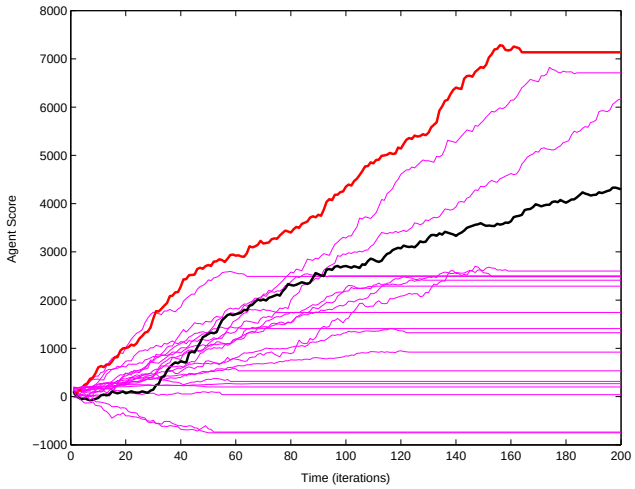


Figure 12.2: Score evolution of agents over one run of the simulation with $N = 21$, $q = 9$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.25$, default type is SGD, $g_{start} = 50$, $g = 10$, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

more realistic than the other is far less obvious and strongly dependent upon what one believes the agent score to represent in terms of a real firm.

For the CTD mechanism, the agent score, $s_i(t)$, could be considered a measure of the total wealth of the firm, i.e., the *value* of the company. When that wealth falls below some level, then the firm is deemed to have defaulted. The growth in the default threshold could reflect a constant rate of inflation.

Or, one might think of the agent score here as the stock price of the firm. If the default growth amount, α, is set to zero, and the original default level, $D(t) = 0$, then the company defaults when its share price hits zero. By using the existing cost, ϵ, and the linking cost, λ, one may calibrate the model to give fair probabilities of default relative to agent size and number of links.

For the SGD mechanism, the agent score may reflect the cumulative sum of income through trading of each firm, after taking account of existing and linking costs. If existing cost is not inclusive of firm debt, than a firm defaults if its income is not sufficient to service its coupon repayments over some period.

12.2 Comparison of New Agent Types

In the CAAM, when agents default, the replacement agents are generated based on a variety of factors. Figure 12.3 shows the evolution of agent scores for a range of new agent parameter values.

In graph (a), the new agents created are all random and the only relation they bear to the defaulted agents is that there is a probability of 0.5 that the maximum bit value, $\mu_{i,max}$ is kept the same rather than being redrawn. In graph (b), the new agents are exact copies of whatever is currently the best performing agent. In graph (c) and graph (f) the new agents are based on the current best performing agent, but with a probability of 0.25 that each strategy bit is redrawn; $q_{best} = 0.25$, and $\mu_{redrawn} = 0.5$. Graphs (d) and (e) show new agents based upon the agent that has just defaulted - an exact copy in (d), and a rough copy ($q_{dead} = 0.25$, and $\mu_{redrawn} = 0.5$) in (e).

As new agents are created, the network will adapt, as some agents are successful and quickly find sufficient neighbours so as not to default. The new agents may also provide competition to existing agents, whether they be exact copies or just have similar strategy vectors. This results in previous strong performers having a lower income from trading and perhaps even defaulting. This is shown in graph (a), where there is a constant turnover of new agents occupying the positions of best performers. In graphs (c) and (f) this turnover is much more rapid and no one agent is the best

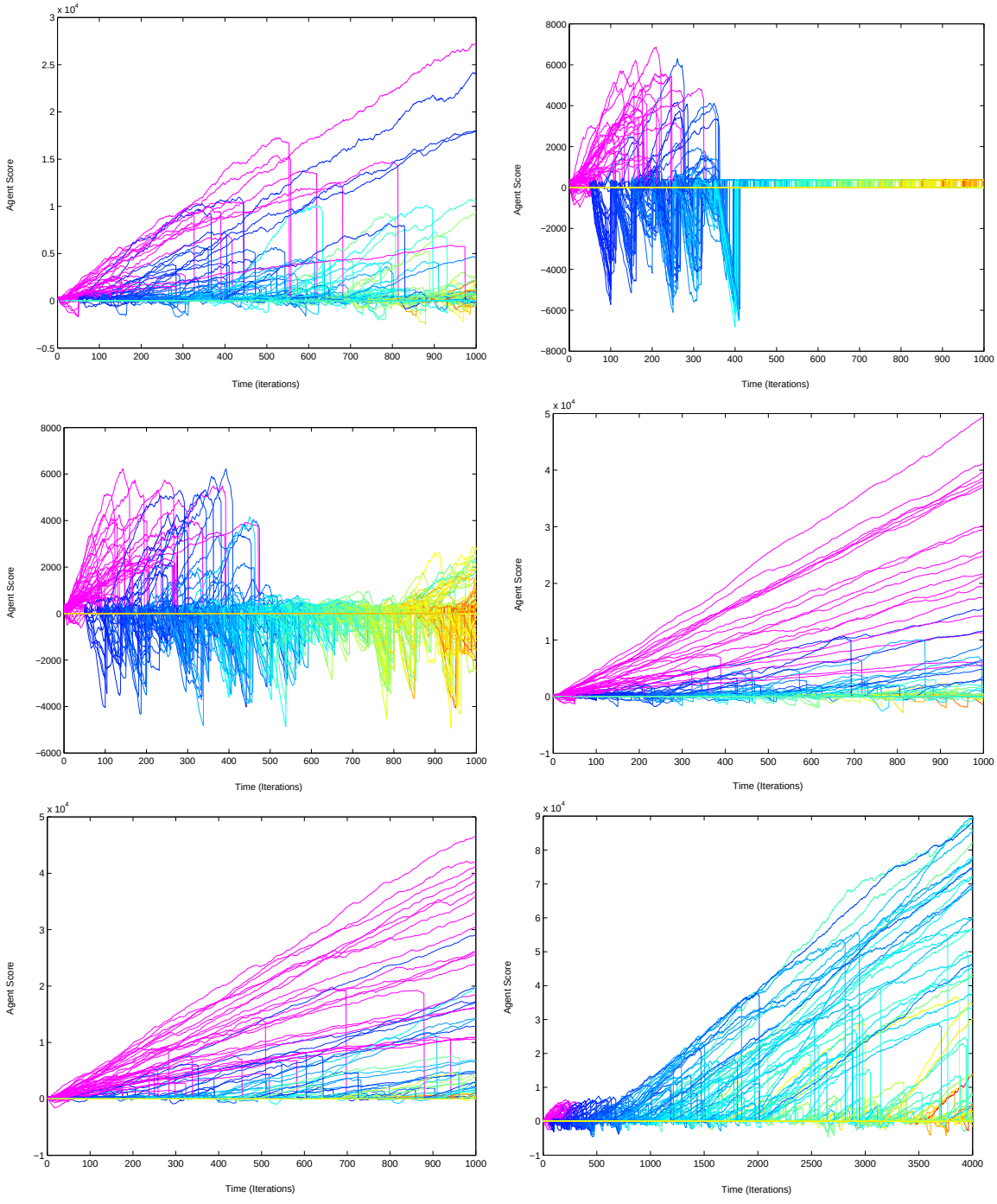


Figure 12.3: Evolution of agent scores for different new agent types. Top left graph (a), top right graph (b), mid left graph (c), mid right graph (d) etc. Common parameters: $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 20$, $g = 10$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 3$, $\kappa_{max} = 20$. (a) $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$, and $\mu_{redrawn} = 0.5$. (b) $r_{rand} = 0.0 : r_{dead} = 0.0 : r_{best} = 1.0$, $\mu_{redrawn} = 0.0$, $g_{size} = 1$, and $q_{best} = 0.0$. (c) $r_{rand} = 0.0 : r_{dead} = 0.0 : r_{best} = 1.0$, $\mu_{redrawn} = 0.5$, $g_{size} = 1$, and $q_{best} = 0.25$. (d) $r_{rand} = 0.0 : r_{dead} = 1.0 : r_{best} = 0.0$, $\mu_{redrawn} = 0.0$, and $q_{dead} = 0.0$. (e) $r_{rand} = 0.0 : r_{dead} = 1.0 : r_{best} = 0.0$, $\mu_{redrawn} = 0.5$, and $q_{dead} = 0.25$. (f) $r_{rand} = 0.0 : r_{dead} = 0.0 : r_{best} = 1.0$, $\mu_{redrawn} = 0.5$, $g_{size} = 1$, and $q_{best} = 0.25$. For colour viewing, 1st generation agents are magenta, then subsequent regenerated agents are blue through to cyan, green, yellow, orange, and finally red.

performer for long. In graph (f) one can see that after an initial period of some 750 iterations, a new group of best performing agents appears that do not default, but rather jostle for position of best agent as new agents are created.

The best copy idea is taken to an extreme in graph (b) with new agents being exact copies of the current best performing agent. After 350 iterations where there is a high turnover of which agent is the best performer, all agents show strong negative score growth and default simultaneously. From this point onwards no agent shows any positive score growth and defaults as soon as its score growth is first assessed. This is because, through time, more and more agents have the same strategies - copies of whichever is the current best performer. Eventually, the strategy set is such that so many of the new agents have identical strategies that there are no agent pairing that will result in a positive score growth.[1].

In contrast to the effect above, by setting the new agents to be copies based on those that have just defaulted, strong performing agents are reasonably unaffected by the new agents' presence in the network. The newly-generated agents are fairly quick to default again - for the same reasons they defaulted initially. However, by comparing graphs (d) and (e) one can see that the *rough* copies in (d) are able to provide more competition to the other agents than the exact copies in (e), and hence more agents with higher scores are defaulting.

12.3 Examining Dependent Defaults

Figure 12.4 shows the evolution of agent scores for 2 runs of the simulation using the CTD default mechanism. The agent highlighted in red can be thought of as a reference agent. Even though there were 21 agents in the simulation, the only other agent scores shown are for agents that were, at some point in the simulation, neighbours to the reference agent, i.e., agents that at some point in the simulation traded with the reference agent. The asterisks indicate when exactly each of these agents was a neighbour.

Graph (a) in Figure 12.4 illustrates the dynamic nature of the agent network - how the reference agent breaks and reforms links with neighbours, perhaps revisiting the same agent many times. One can also see that around iteration 70 a new link to an agent (whose score is just below that of the reference agent at that time) results in a large rally in the reference agent's score, $s_i(t)$. This new link was possible due to

[1]This "collapse" in the system is dependent upon the time before each agents' score growth is assessed, g_{start}, and the time period over which the growth is assessed, g. For high values of g_{start} and g this effect has been shown not to appear even after a long time (see plot (b) figure 12.14)

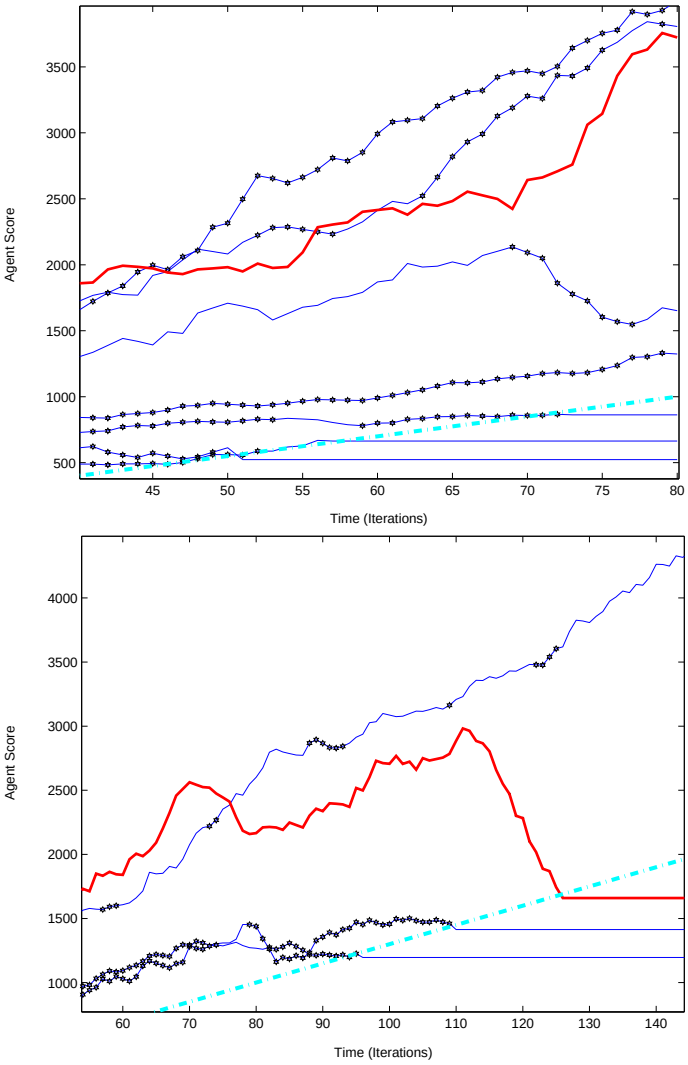


Figure 12.4: Score evolution of agents over two runs of the simulation ((a) and (b)) for same parameters; $N = 21$, $q = 9$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.25$, default type is CTD, $D(0) = -200$, $\alpha = 15$, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

the default of an agent previously attached to the reference agent's new neighbour. This illustrates a positive default dependence between agents - one agent benefiting from another agent's default.

Graph (b) illustrates contagious default. The reference agent here quickly defaults once 2 of its neighbours have defaulted. This demonstrates negative default dependence and default contagion within the network.

Both positive and negative default dependence, and hence default contagion, are certainly effects one would expect to observe in the real world. For example, negative dependence might result if a tyre firm's major client (a large car manufacturer) were to default. Positive dependence might result if one of a number of competitors in an oligopoly (say, the Trans-Atlantic airline industry) were to default leaving a larger market share to those companies that remain.

12.4 Agent Network Degree Distributions

A degree distribution is a histogram of the number of links an agent possesses (degree of each node) versus the number of agents of the total active population with that degree.

Figure 12.5 shows the evolution of the degree distributions for the regular modified and CAAM versions of the agent network model, with accompanying agent score plots. From both degree distributions one can see the stochasticity of the network structure resulting from agents changing sets of neighbours as they try to optimize their payoffs from trading.

In graph (a) the average degree is shown to reduce as the number of active agents goes down. Trivially, this is because fewer active agents results in fewer potential neighbours that can combine to form each agent's optimal set.

Graph (b) refers to a single run of the CAAM. The degree distribution does vary from iteration to iteration - partially as a result of agents having more than one optimal set of neighbours from which they randomly choose one - but mainly this will be due to the default of some agents and the introduction of random new ones. Although agents are frequently altering their set of neighbours, the shape of the overall distribution does appear to be roughly constant through time.

Figure 12.6 shows the average degree distributions for increasing numbers of agent technologies (or strategy bits). It is clear from the graphs that the average degree increases as the number of strategy bits is increased. This change is marked when

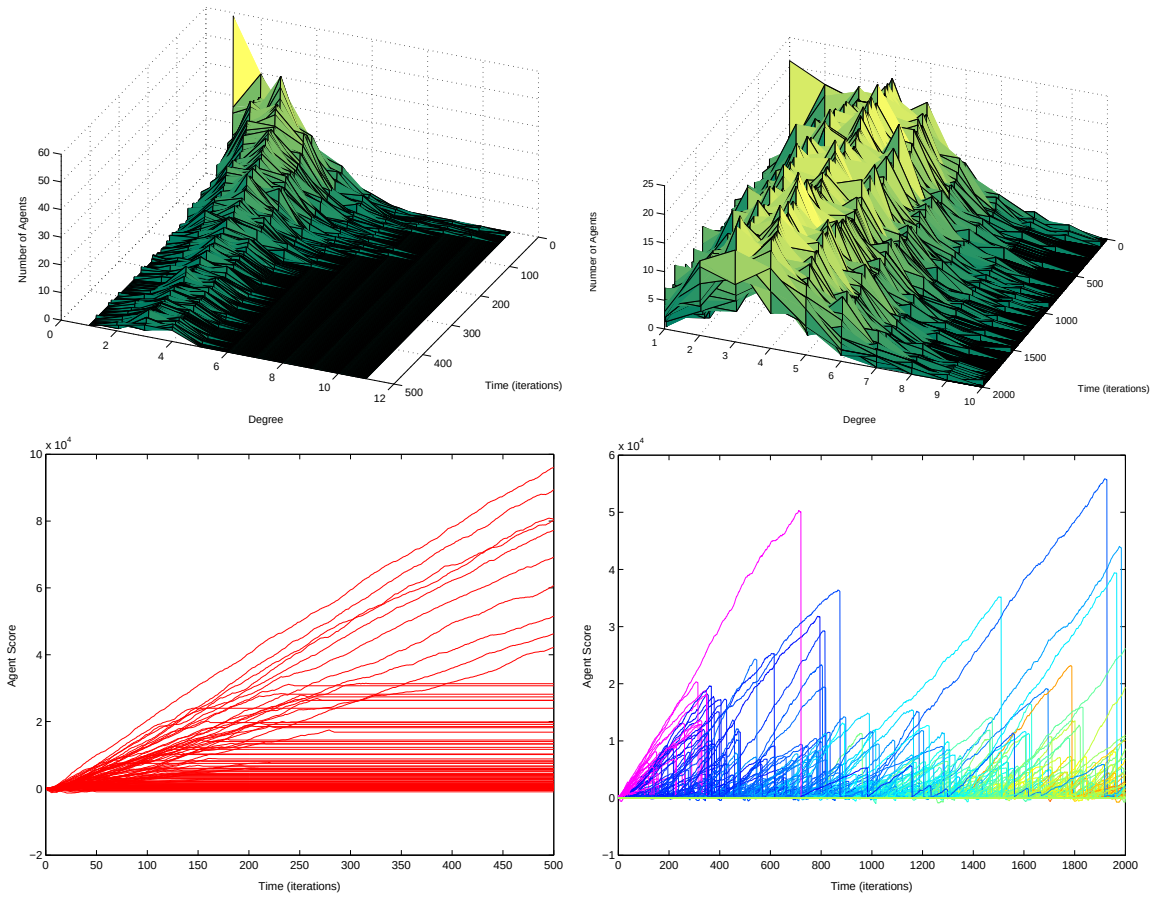


Figure 12.5: Evolution of degree distributions through time with accompanying plots of agent scores for regular modified agent network model and CAAM version respectively. For left-hand plots, $N = 100$, $q = 9$, $s_i(0) = 100$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 50$, $g = 10$, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$. For right-hand plots, $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 20$, $g = 10$, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$, and $\mu_{redrawn} = 0.5$. For colour viewing, in bottom right plot, 1st generation agents are magenta, then subsequent regenerated agents are blue through to cyan, green, yellow, orange, and finally red.

comparing the 3-bit and 15-bit ((a) and (e)) distributions where the modal degrees are 1 and 4 respectively.

With respect to real firm-firm interaction, one would expect that if the number of different technologies in which each firm traded were to increase, then so would the number of business links each firm would need to increase their wealth or share price. The same logic applies here to the agent network model. As the number of strategy bits is increased, one would expect that on average each firm would need more links to neighbours to maximize their potential payoff.

12.5 Agent Lifetimes

Here we study the lifetimes of agents in the CAAM. Figure 12.7 shows the survival probability versus agent lifetime for a single run over 400,000 iterations with random new agent types.

For lifetimes greater than 100 iterations there is an approximate power-law relationship between the survival probability and the lifetime. Therefore, in this region, the survival probability can be approximately written as:

$$P(\tau > t) \propto t^{\varphi}, \tag{12.1}$$

where τ is the default time and hence $P(\tau > t)$ is the survival probability of the agents. The exponent $\varphi \approx -0.7$. (For measures of the reliability of fitting to the power-law distribution refer to [42].)

For the CAAM the lifetimes are approximately exponentially distributed with power-law tails. The change from approximately exponential to power-law distribution may be the result of effectively looking at two types of agent. As agents are given $g_{start} = 50$ iterations before their performance is first assessed, new agents that only survive for less that 100 iterations will not have had the chance (or perhaps been able) to form any substantial and rewarding links to the existing network. Hence, these new agents default nearly as soon as their performance is measured and form the exponential distribution of lifetimes. Agents surviving for more than 100 iterations will have formed enough links to support themselves and become integral constituents of the agent network. The default of these more successful agents forms the power-law tails.

The fat tails indicate that a greater proportion of agents live for a long time than would be the case if the lifetime distribution were wholly exponential. Power-laws are common in systems of interacting agents and are indicative of the cooperative

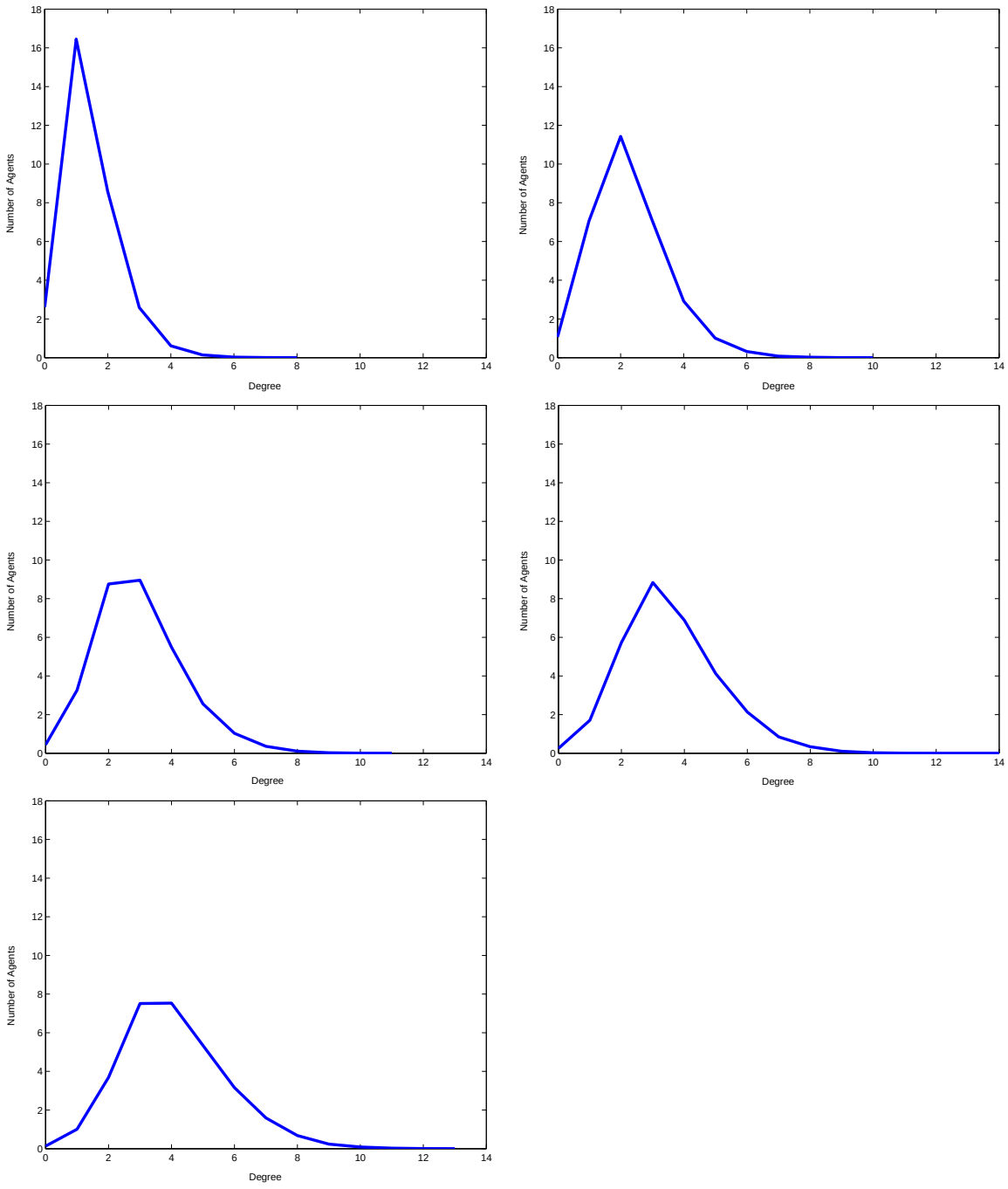


Figure 12.6: Degree distributions after 50 iterations averaged over 2000 runs of simulation for different numbers of technologies or strategy bits. Uses CTD default mechanism with parameters such that no default is possible. Graphs are labelled; (a) top left, (b) top right, (c) mid left, etc. Common parameters: $N = 31$, $s_i(0) = 100$, $z = 0.25$, $\beta = 0.75$, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$. (a) $q = 3$. (b) $q = 6$. (c) $q = 9$. (d) $q = 12$. (e) $q = 15$.

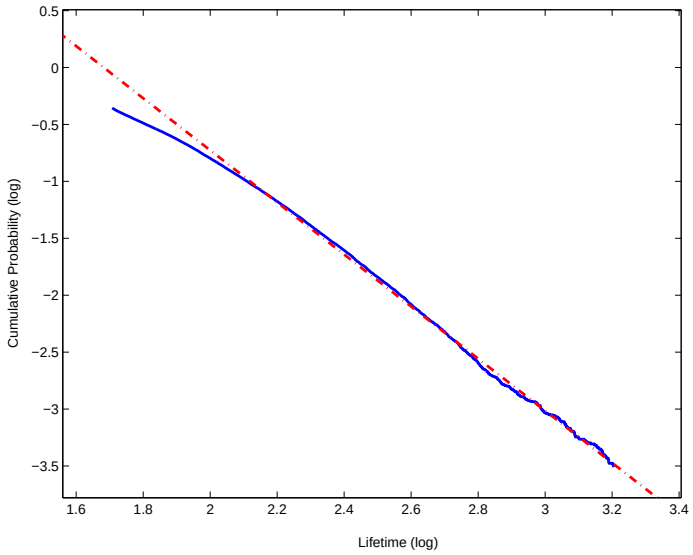


Figure 12.7: Log-log plot of probability of agent survival versus agent lifetime, using CAAM with randomly drawn new agents. One run of simulation over 400,000 iterations resulting in 35,000 defaults. $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 50$, $g = 40$, $\epsilon = 1.0$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$, and $\mu_{redrawn} = 1.0$.

behaviour amongst the system's constituents [43]. Hence, the appearance of power-law behaviour here is not unexpected.

In contrast, an agent-based model by Axtell, in which heterogenous agents group together to form firms indicates a predominantly exponential distribution of firm lifetimes [44].

However, it is unclear exactly how real world firm lifetimes are distributed. Using data from real world firms it is very difficult to provide a good measure of firm lifetimes due to changes in firm structure and ownership resulting from mergers, takeovers, acquisitions etc. These effects are not considered in our model nor that of Axtell. However, a study by Fujiwara into the defaults of over 16,000 Japanese firms in 1997 showed the distribution of firm lifetimes to be exponential [41].

Therefore, due to the relative lack of empirical studies and the inherent difficulty in measuring firm lifetimes it is unclear, up to now, whether our model is good representation of the real world.

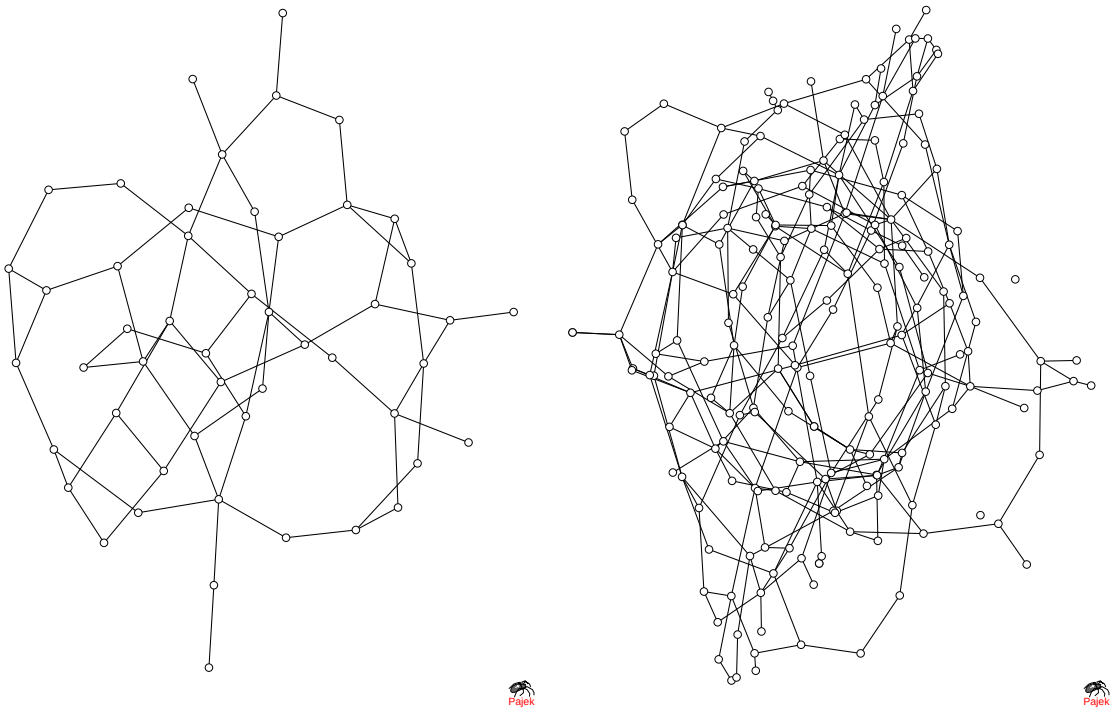


Figure 12.8: Plots of agent network using Pajek for $N = 50$ and $N = 200$. Common parameters; $q = 9$, $z = 0.25$, $\beta = 0.75$, $\kappa_{min} = 1$, $\kappa_{max} = 50$, $\epsilon = 0.5$, and $\lambda = 1$. Uses CTD default mechanism with parameters such that no default is possible.

12.6 Visualising the Agent Networks

The adjacency matrix $\underline{\underline{A_t}}$ can be represented graphically using a network drawing package such as Pajek [45]. Figure 12.8 and Figure 12.9 show typical network structures for 50, 200 and 500 agents respectively. The agents are represented as vertices and the edges between vertices represent that a trade link exists, at that point in time, between those two agents, i.e., for two agents, i and j, a trading link exists between them, and hence an edge is drawn on the network plot, if $\underline{\underline{A_t(i,j)}} = \underline{\underline{A_t(j,i)}} = 1$. In the plots, all edge weighting is the same and is not indicative of the trading link. Note how all graphs show the presence of large loops and a large number of agents with greater than one link.

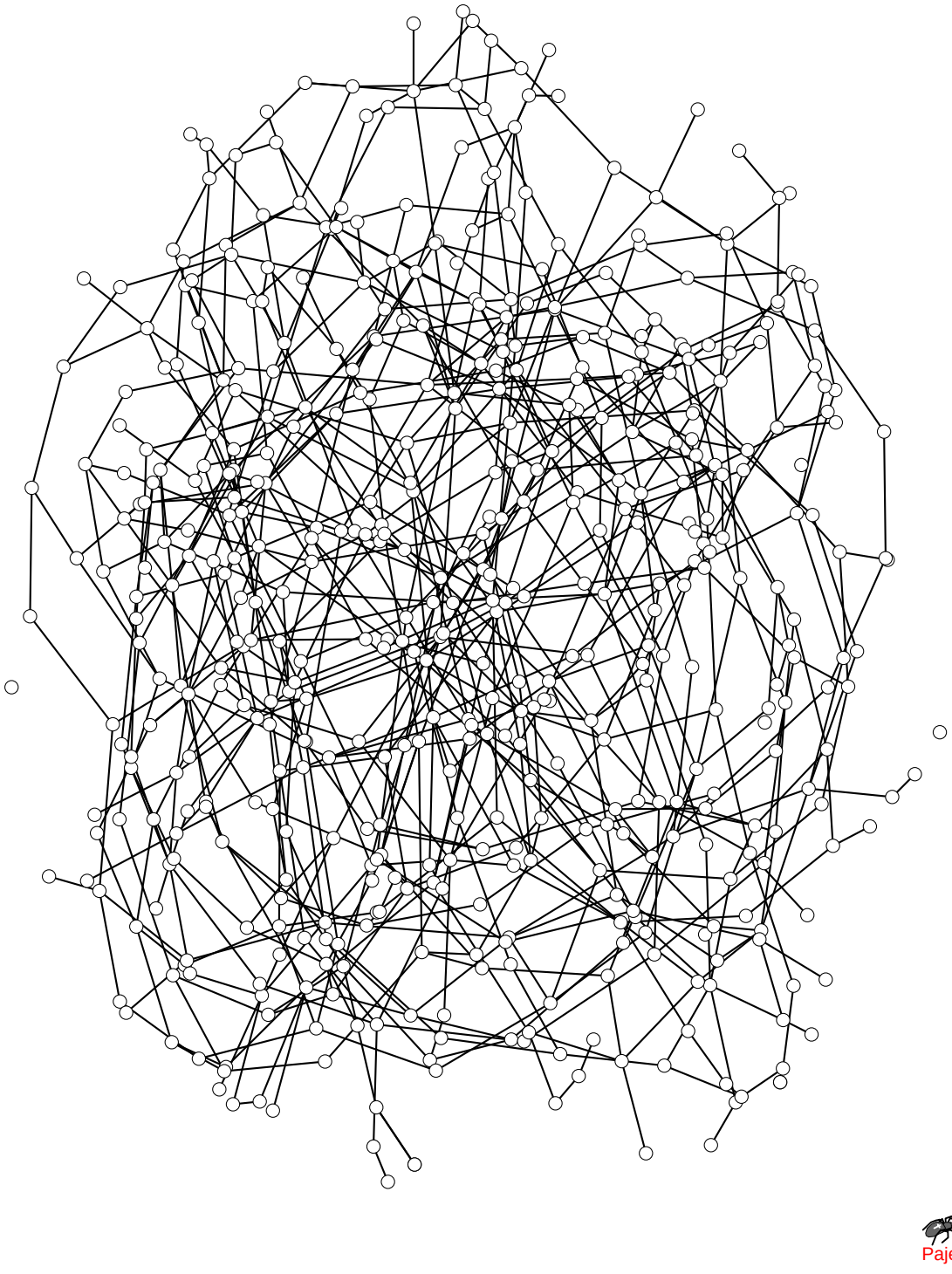


Figure 12.9: Plot of agent network using Pajek for $N = 500$, $q = 9$, $z = 0.25$, $\beta = 0.75$, $\kappa_{min} = 1$, $\kappa_{max} = 50$, $\epsilon = 0.5$, and $\lambda = 1$. Uses CTD default mechanism with parameters such that no default is possible.

12.7 Are Agent Networks Small-world?

Using the Watts definition of small-world graphs we are able to test whether the networks generated by the interacting agents in our model are small world [46]. The Watts definition of small-world is as follows:

> Relational graphs admit a particular class of graphs that exhibit characteristic path lengths approximately the same as equivalent random graphs, but with much greater clustering. These graphs are called *small-world graphs*.

As some of the agents are not connected to the network, i.e. the graph is not *complete*, it is necessary to use the "harmonic mean" geodesic distance, l, to measure the path length between all agent pairs [47]. This is given by:

$$l^{-1} = \frac{1}{\frac{1}{2}N(N+1)} \sum_{i \geq j} d_{ij}^{-1}, \tag{12.2}$$

where i and j are any two vertices, N is the total number of vertices (or active agents) and d_{ij} is the geodesic distance from vertex i to vertex j. Using the harmonic mean means that infinite values of d_{ij} between unconnected vertices contributes nothing to the sum.

The neighbourhood of v is the subgraph made up of all the neighbours of v, not including v itself. The clustering coefficient, γ_v, for the neighbourhood of a vertex, v, is given by Watts and defined as:

$$\gamma_v = \frac{|E(\Gamma_v)|}{(\frac{k_v}{2})}, \tag{12.3}$$

where Γ_v is the neighbourhood of vertex v, $|E(\Gamma_v)|$ is the number of edges in the neighbourhood of v, k_v is the number of vertices in the neighbourhood of v, and $(\frac{k_v}{2})$ is the number of *possible* edges in Γ_v. One can think of γ_v as the net fraction of possible edges that actually occur in Γ_v, or the probability that two vertices in Γ_v will be connected.

Alternatively, but equivalently, one may prefer to consider the clustering coefficient for the neighbourhood centred on vertex v as:

$$\gamma_v = \frac{\text{number of triangles connected to vertex } v}{\text{number of triples centred on vertex } v}, \tag{12.4}$$

where the difference between a triple and a triangle is illustrated in Figure 12.10.

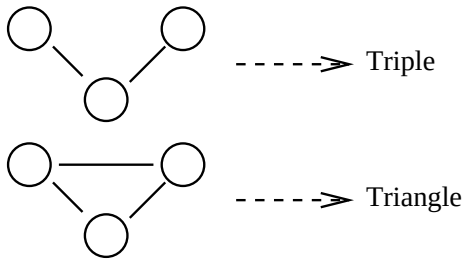


Figure 12.10: Difference between a triple and a triangle.

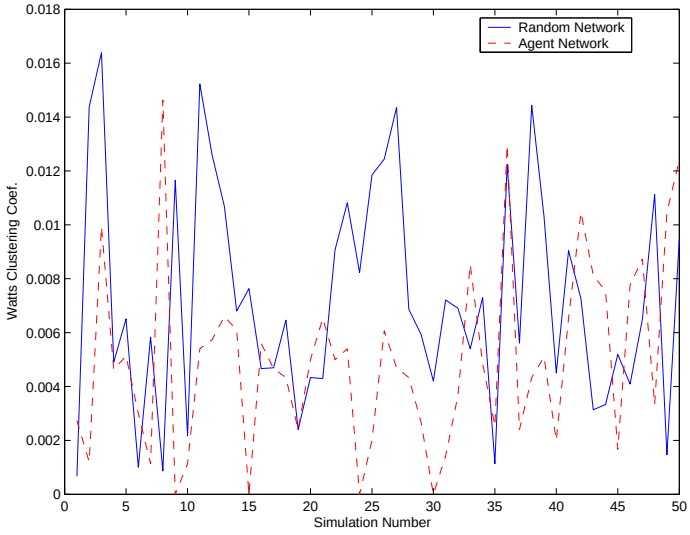


Figure 12.11: Watts clustering coefficient for agent network (evolved over 1000 iterations) and comparable random network, for 50 realisations of each network, $N = 500$, $q = 9$, $z = 0.25$, $\beta = 0.25$, no default possible, $\epsilon = 0.5$, $\lambda = 1.0$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

Then, the clustering coefficient for the whole network, γ, is γ_v averaged over all vertices v.

Figures 12.11 and 12.12 show the clustering coefficient and harmonic mean geodesic distance respectively for 50 different agent networks and comparable random networks. A random network is created by selecting vertex pairs at random and building an edge between those two vertices if one does not already exist. A random network was considered comparable if it had the same number of vertices and links as the agent network.

Referring to Figure 12.11, the clustering coefficient for the agent network is of the same magnitude as that of the random graph. In fact, in comparison to many other networks the level of clustering is very low [47]. Most social, information, and biological networks give much higher values for the clustering coefficient with only a

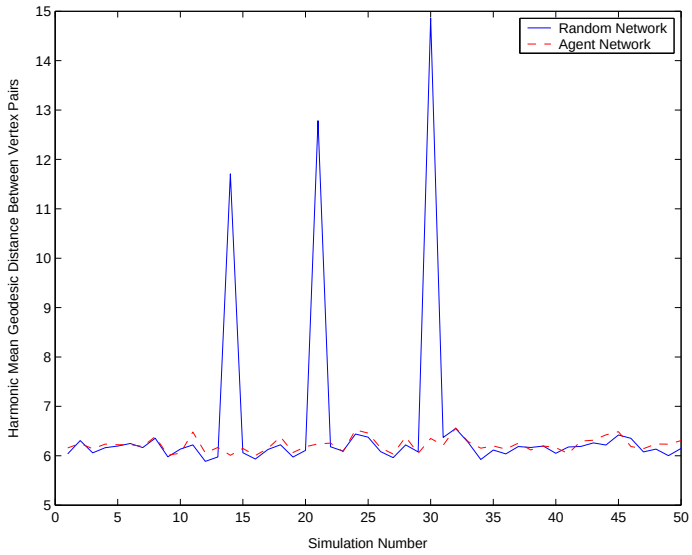


Figure 12.12: Harmonic mean geodesic distance for agent network (evolved over 1000 iterations) and comparable random network for 50 realisations of each network, $N = 500$, $q = 9$, $z = 0.25$, $\beta = 0.25$, no default possible, $\epsilon = 0.5$, $\lambda = 1.0$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

few networks including electronic circuits and software classes having similar values.

The mean geodesic path lengths in Figure 12.12 for the agent and random networks are very similar bar a few outlying values for the random network.[2]

Therefore, the agent networks are not small-world as defined by Watts. The reason for this is illustrated in Figure 12.13. In this very simple example agent B is linked to agent C and agent A - i.e., A and C make up the neighbourhood of B. The strategy bits can only take values of ± 1. The maximum potential score for each link is shown. Note how there is only one score for each link as the removal of any zero bits removes the asymmetry in agent pairing.

There are no strategy vectors R_A, R_B, and R_C for the parameters above such that the score for each and every link, A to B, B to C, and C to A, is greater than zero. Therefore, for $q = 3$ it is impossible for a triangle, and hence a cluster, to form in the network. This is also the case for $0 \leq q \leq 4$.

Although this is a very simple illustration, the idea translates to the more general case. As an agent will try to link to other agents with anti-correlated strategies so as

[2]In fact, the mean vertex-vertex distance, l, for both random and agent networks is 6, which is coincidently the same value of mean path length measured by Stanley Milgram in a social science study in the 1960's, which in effect coined the phrases "six degrees of separation" and "small-world" [48].

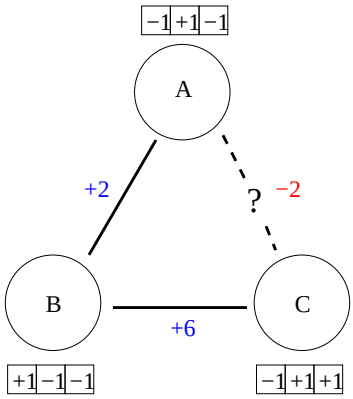


Figure 12.13: Simple case illustration for low level of clustering in agent networks. Agent B is connected to C and A. A is considering linking to C.

to maximize their potential payoff, it is therefore unlikely, although not impossible in general, that these neighbour agents will want to link to each other. The logic is the opposite of social networks where if a person is friends with two others, then there is a high chance that they too will also be friends [49].

The lack of triangles can be seen in the network visualisations in Figures 12.8 and 12.9. The graph for $N = 50$ does show very few triangles but a large proportion of squares, which fits with the theory above.

12.8 Agent Learning.

Figure 12.14 shows the change in the time between successive default events for different new agent types in the CAAM.

One can see that the only new agent type to show a significant increase in the time between consecutive default events are new agents that are rough copies of the current best performing agents.

This would indicate that the system of agents that are rough copies of the current best performing agents is adapting such that, on average, each agent defaults less often. This system of agents is demonstrating a Darwinian "survival of the fittest" mechanism. If the new agents were exact copies of the best agent (graph (b)), there is no increase in the time between consecutive default events (even after 18,000 defaults). However, by including a form of strategy vector mutation between generations of agents, the system is able to adapt to increase the probability of agent survival.

Even though for the random new agent types (graph (a)) and the rough copies based on the recently defaulted agents there is also a change to the strategy vector

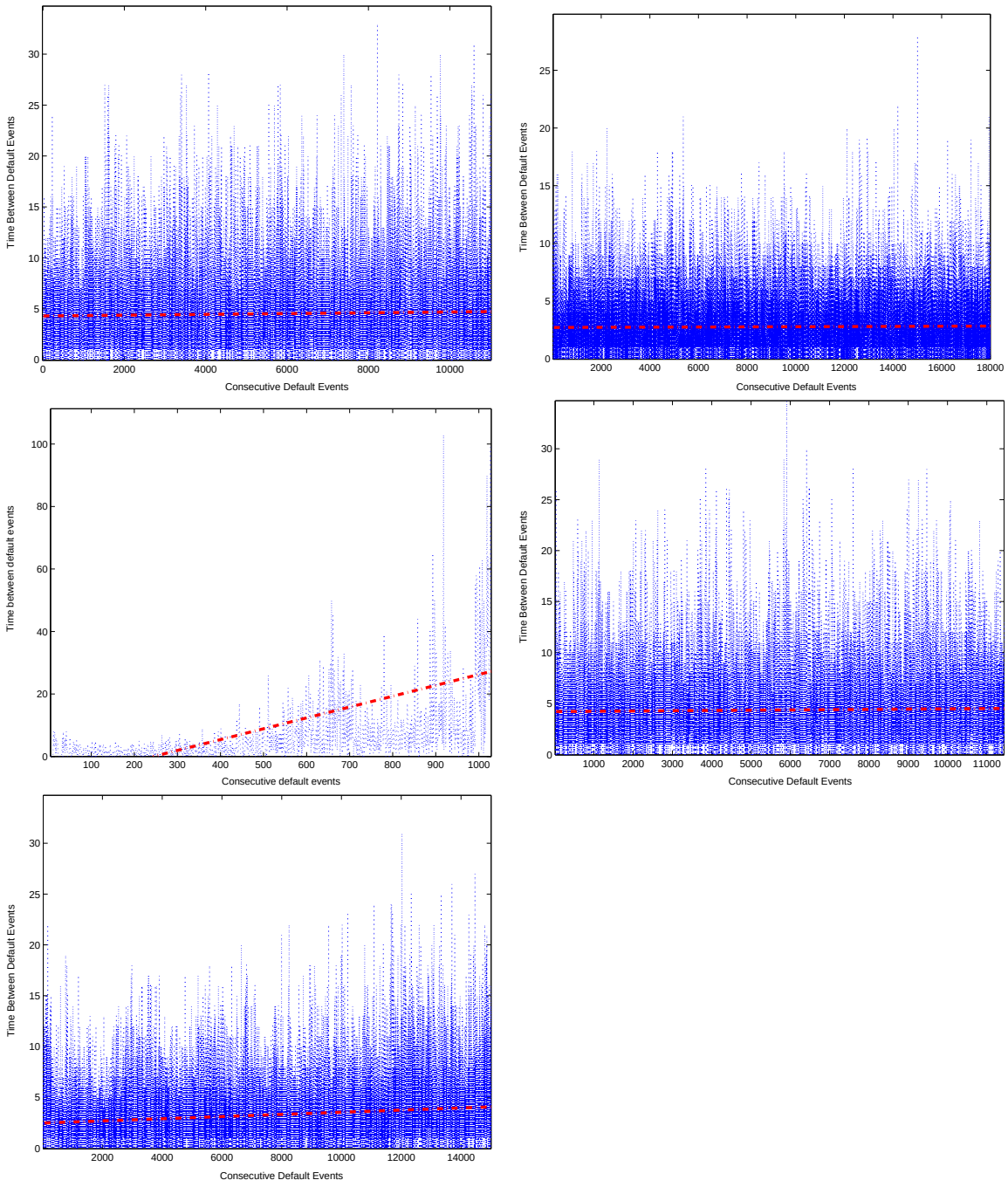


Figure 12.14: Plots of time between default events versus consecutive default events over 50,000 iterations. Bold dashed line indicates linear fit. Plots are labelled; (a) top left, (b) top right, (c) mid left, (d) mid right, etc.Common parameters: $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 50$, $g = 40$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$. (a) $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$, and $\mu_{redrawn} = 0.5$. (b) $r_{rand} = 0.0 : r_{dead} = 0.0 : r_{best} = 1.0$, $\mu_{redrawn} = 0.0$, $g_{size} = 1$, and $q_{best} = 0.0$. (c) $r_{rand} = 0.0 : r_{dead} = 0.0 : r_{best} = 1.0$, $\mu_{redrawn} = 0.5$, $g_{size} = 1$, and $q_{best} = 0.25$. (d) $r_{rand} = 0.0 : r_{dead} = 1.0 : r_{best} = 0.0$, $\mu_{redrawn} = 0.0$, and $q_{dead} = 0.0$. (e) $r_{rand} = 0.0 : r_{dead} = 1.0 : r_{best} = 0.0$, $\mu_{redrawn} = 0.5$, and $q_{dead} = 0.25$.

between successive generations of agents, there is still only a slight increase in time between consecutive default events. It appears that there needs to be a combination of copying the best agents *and* altering the strategy vector to increase the average lifetime of each agent.

By considering the results for this Section and for section 12.2 it appears that to promote a healthy overall marketplace for all firms whilst still ensuring the best chance for each individual new firm, that new firms should be based upon *successful* firms that already exist. Instead, they should differ in some way to each of the successful firms upon which they are based so as to maximize both their own chances of success and also assist the whole market by increasing the average lifetime of all firms.

12.9 Evidence of Clustered Defaults

One of the so-called "stylised facts" of credit defaults is that default events appear to be clustered together through time. Although, as has been mentioned before, empirical studies into credit events are relatively scarce (partially because of the lack of data and classification of exactly what constitutes a default event), a study by Ormerod and Smith in 2001 into the survival of the world's largest companies in 1912 shows an approximate power-law between the frequency of annual extinction rates [43]. From 1912 until 1995, 50 of the 100 companies had defaulted. Of those 50 defaults, there were 53 years in which no defaults occurred and one year when as many as 6 defaults occurred. Drawing conclusions from any study based on default events should be taken with extreme caution, as defaults happen very infrequently and hence provide very few data points upon which analysis can be made. However, the results in the Ormerod paper would imply a clustering nature to the default events.

One argument for clustering of defaults is the cyclical boom-bust behaviour of many economies with all firms being subjected to the same external market conditions such as those of a recession. Another factor is the default dependence between companies due to their trading or business links. This default dependence has been shown to exist in our agent model in Section 12.3. Therefore, it may be reasonable to expect the agent based model for credit default to demonstrate default clustering.

Measuring default clustering in the agent model can be approached in a variety of ways. Detailed below are two different approaches, firstly looking a histograms of

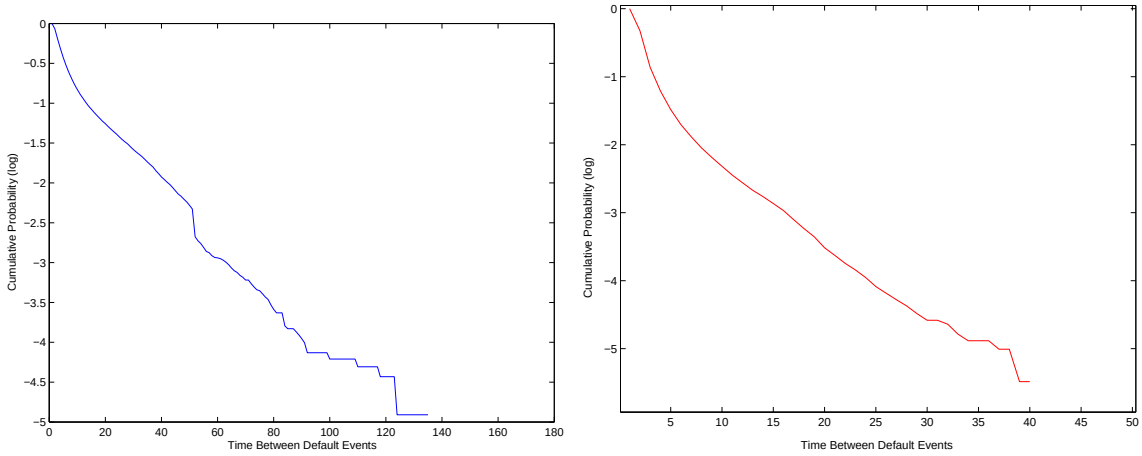


Figure 12.15: Plots of log of survival probability of all active agents versus time between default events for the two alternative default mechanisms for CAAM. Common parameters: $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$, and $\mu_{redrawn} = 1.0$. (a) Default type is SGD, $g_{start} = 50$, and $g = 40$, and the number of default events is $\approx 81,000$. (b) Default type is CTD, $D(0) = -2000$, and $\alpha = 20$, and the number of default events is $\approx 300,000$.

number of default events versus time between default events, and secondly considering the auto-correlation of default events.

We can denote τ_{next} as the time between default events of all agents combined, i.e., the time between the defaults of any of the agents within the network. Figure 12.15 shows plots of the log of the probability, $P(\tau_{next} \geq t)$ against time t, where t is the time between default events. This is analogous to the survival probability measure in Section 12.5. The first (left-hand) graph is for the SGD default mechanism and the second graph relates to the CTD default mechanism. If default events were purely random then the plots would show straight lines - there would be a exponential distribution of default events. Therefore, default events for this model could be represented as a Poisson process, i.e., one could assume that each of the default events is independent and unconditional on the time of the previous event.

The results in the first plot in Figure 12.15 show that the distribution of default events under the score growth default mechanism is not exponential. Note, at $t = 50$ there is a clear jump. One would assume this is the result of the SGD default mechanism and coincides with the time after creation at which a new agent's performance is first assessed, $g_{start} = 50$. The jump relates to periods in the simulation run in which a new agent is created and then defaults straight away before any other agent in the

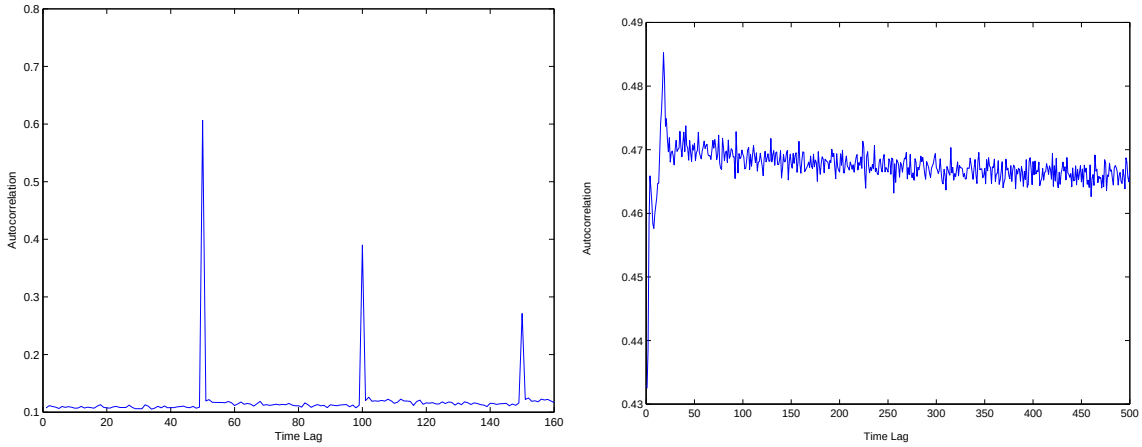


Figure 12.16: Plots of auto-correlation of number default events per iteration versus time lag for CAAM. Common parameters: $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$, and $\mu_{redrawn} = 1.0$. (a) Default type is SGD, $g_{start} = 50$, and $g = 40$. Number of default events is $\approx 81,000$. (b) Default type is CTD, $D(0) = -2000$, and $\alpha = 20$. Number of default events is $\approx 300,000$.

group default. As the majority of new agents will defaults after g_{start} iterations, there is a sudden drop in the probability at $t = 50$.

The second plot in Figure 12.15 again illustrates a non-exponential nature of the distribution of default events. Also, the lack of a jump in the cumulative probability for the second plot confirms the jump in the first is a result of the SGD default mechanism.

One may tentatively conclude from these results that the default events are not occurring purely at random, which one may expect from the complex nature of the interactions between the agents in the network and the default dependence already demonstrated.

Figure 12.16 shows two auto-correlation plots versus time lag, the first for the SGD default mechanism, the second for the CTD default mechanism. The auto-correlation measure , $r(k)$, is given as follows:

$$r_k = \frac{\sum_{i=1}^{N_{it}-k}(Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N_{it}}(Y_i - \bar{Y})^2}, \tag{12.5}$$

where Y_i is the number of defaults at iteration i, $\bar{Y}$ is the mean number of defaults per iteration, k is the time lag, and N_{it} is the total number of iterations over which the measurement is made.

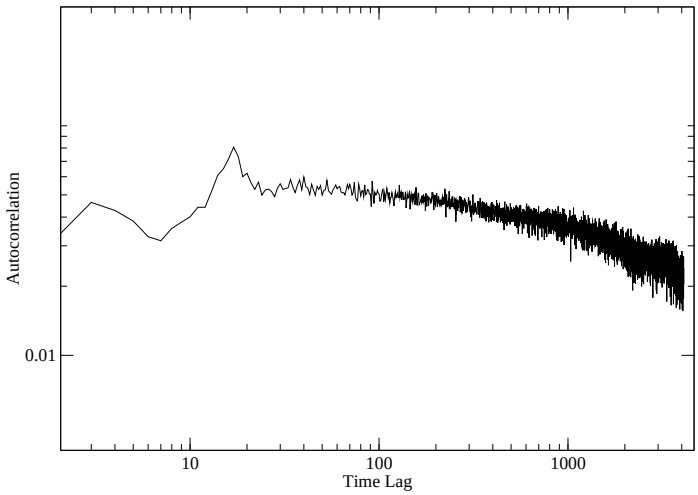


Figure 12.17: Log-log plot of auto-correlation of default events versus time lag, $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is CTD, $D(0) = -2000$, $\alpha = 20$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$ and $\mu_{redrawn} = 1.0$.

By comparing the two plots in Figure 12.16, one can observe the marked effect on the auto-correlation measure of changing the default mechanism. The first plot (SGD mechanism) consists of a large spike at 50 iterations and a smaller spikes at 100 and 150 iterations. Again, this is because of the time waited before first assessing the performance of a new agent, $g_{start} = 50$. Because of the high proportion of agents that fail as soon as they are assessed, the peaks are a result of a high level of auto-correlation between successive generations of the same agent. The reason for the decrease in the size of the peak with increasing time lag is that with each further generation of agent there is a probability that it will not fail straight away and go on to exist for more that 50 iterations.

The second plot in Figure 12.16 relates to the CTD default mechanism. Here we see a sharp rise in the auto-correlation to around 0.47 and then a very gradual drift down - even after a lag of 500 iterations the auto-correlation is only slightly lower than it is at 50 iterations. This is consistent with a time series that incorporates some overall trend, such as a gradual rise or decline the in value being measured, and hence, a non-constant rolling mean. The time series under consideration here - the number of defaults per iteration - is dominated by a slow increase in the average time between default events which is a result of the agents evolving so that their rate of default reduces with time (see Section 12.8 on agent learning).

One can approximately remove the effect of the agent learning on the time series

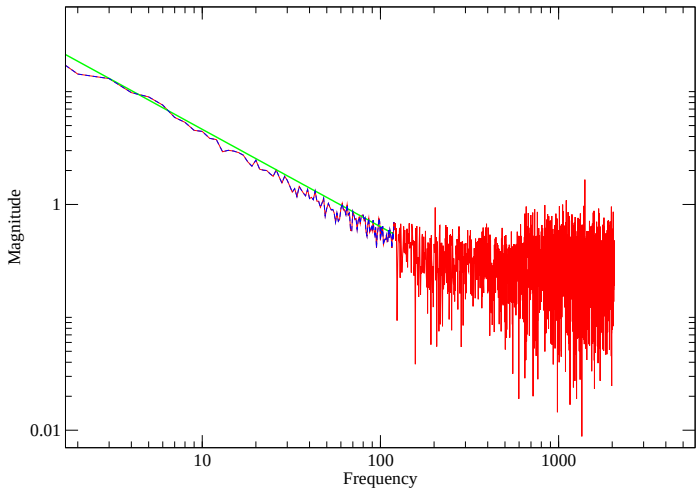


Figure 12.18: Log-log plot of frequency spectrum of auto-correlation of default events versus time lag. The data highlighted in blue is a subset of all the data, and the green line is a power-law fit to this data, $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is CTD, $D(0) = -2000$, $\alpha = 20$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$ and $\mu_{redrawn} = 1.0$.

used to produce the righthand plot in Figure 12.16 for the CTD default mechanism. This is done by letting $\bar{Y}$, the mean number of default events per iteration, given in equation 12.5, vary as a function of time, i.e., rather than calculate $\bar{Y}$ for the whole data series and then calculate the auto-correlation, a cubic fit can be made to the number of defaults per iteration and the values given by this cubic fit used instead of the mean number of defaults per iteration.[3] Figures 12.17 and 12.18 are generated using a de-trended time series of time between default events using the method given above. Figure 12.17 shows the autocorrelation plotted on a log-log plot. Note that now the auto-correlation remains low and slowly approaches zero. The frequency spectrum of this data is given figure 12.18. For lower frequency levels we see a clear power-law decay of the spectrum as a function of frequency, which translates to a power-law decay for the correlations as a function of time. For the higher frequency levels (shown in red) the spectrum appears flat. This is a region of white noise, where all frequencies are represented with equal weights corresponding to uncorrelated increments. The power-law decay implies a long-memory process for the default events. This is again not unexpected when one considers the default dependence between agents. The default mechanisms themselves are based around

[3]This is equivalent to subtracting from the number of defaults per iteration, the values given by a cubic fit to this data and then calculating the auto-correlation on this modified time series.

the agent score $S_i(t)$ and an agent will take time to reach the criteria for default given the default of neighbours upon which it depended for survival.

We have shown that the distribution of default events is not exponential and that the autocorrelation of default events demonstrates that defaults are governed by a long-memory process. Whether this is satisfactory evidence for significant clustering in the defaults is unclear, however, it does indicate that the defaults are non-random and implies a non-trivial dependence between all agents.

12.10 Hurst Exponent of Agent Score Evolution

We define the Hurst function of the evolution of an agent's score to be:

$$\mathcal{H}(\tau) = \sqrt{\langle (s_i(t) - s_i(t+\tau))^2 \rangle}, \tag{12.6}$$

where $s_i(t)$ is the score of agent i at time t, τ is the time lag, and $\langle ....\rangle$ is the average over time. The Hurst exponent, H, is defined from

$$\mathcal{H}(\tau) \propto \tau^H, \tag{12.7}$$

and can be thought of as a way to characterize the temporal development of the agent's score fluctuations [51]. It is a measure of the scalability of the process with time.

For a normally distributed random walk the Hurst exponent is $H = \frac{1}{2}$. A Hurst exponent in the range $0.5 < H < 1.0$ represents a long-memory process and is the result of persistent behaviour, and hence some underlying predictability to the agent's score evolution. In Figure 12.19 we see the Hurst function for one agent chosen at random in a simulation in which no agent can default. The red dashed line is a power-law fit for low values of τ and gives a Hurst exponent, $H = \frac{1}{2}$. This result is expected, as for a long run of the simulation in which no agent can default the network should remain relatively stable. Therefore, the predominant factor in the score process is the random sampling of the agent strategy vector R_i, given by the parameter β. Hence, under these conditions, over a large number of iterations, the central limit theorem implies that resulting score process can be approximated as a normally distributed random walk.

It has been shown that the Hurst exponent for a selection of liquid assets including the S&P 500 and the long-term German interest rate, have typical values of $H = \frac{2}{3}$ for low τ [50]. As τ increases, the Hurst exponent for liquid assets approaches $\frac{1}{2}$, i.e., the same as that of a normally distribution random walk. In comparison with

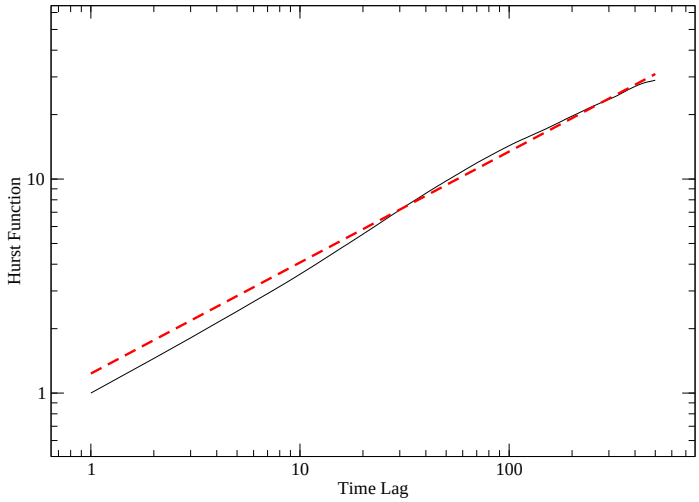


Figure 12.19: Log-log plot of the Hurst function versus time lag for the normalised de-trended evolution of the score of an agent. The dashed red line indicates a power-law regression in the range $1 < \tau < 500$. The score was for an agent run over 10,000 iterations with $N = 11$, $q = 9$, $z = 0.25$, $\beta = 0.25$, no default possible, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

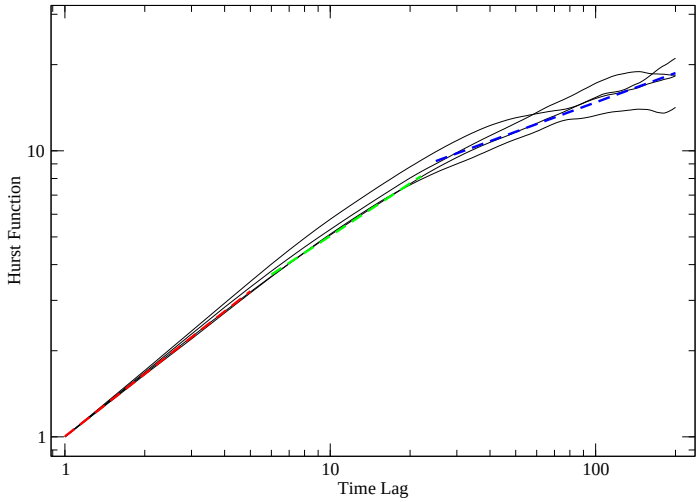


Figure 12.20: Log-log plot of the Hurst function versus time lag for the normalised de-trended evolution of the score of 4 different agents from the same simulation. The dashed red line indicates a power-law regression in the range $1 < \tau < 5$, the green line for range $6 < \tau < 22$, and the blue line for the range $25 < \tau < 200$, for the longest surviving agent. The score was for an agents runs ranging between 1378 and 3129 iterations with $N = 51$, $q = 11$, $s_i(0) = 400$, $z = 0.25$, $\beta = 0.75$, default type is SGD, $g_{start} = 50$, $g = 40$, $\epsilon = 1$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 20$, $r_{rand} = 1.0 : r_{dead} = 0.0 : r_{best} = 0.0$ and $\mu_{redrawn} = 0.5$.

Figure 12.20, Figure 12.20 shows the Hurst function for the CAAM where the score of four of the longest living agents was studied. Here, unlike above, these agents would have been subject to the effects of the defaults of many other agents within their lifetimes. The Hurst exponents are measured for low, medium, and high τ values and are approximately 0.7, 0.6, and 0.3 respectively, although the high τ values should be viewed with caution as they are subject to large fluctuations in the Hurst function as shown in the figure.

The medium and low τ values are higher than those of the simulation in which there were no defaults occurred, and hence higher than those of a normally distributed random walk. This implies that the default of other agents and thence the continual reordering of the agent network results in a persistence within the score process and therefore some level of predictability. The values of the Hurst exponent approximately match those of the real world indices, as does the trend of moving from a higher to lower exponent with increasing τ.

12.11 Measuring Agent Dependence

When an agent defaults, it can have either a positive or negative effect upon the other remaining agents in the network. A measure of this dependence is to study the change in agent score, $s_i(t)$, given the default of another agent. Figure 12.21 shows the score evolution for a selection of 4 agents all in the same network. After 1000 iterations, one agent is removed from the simulation. (The simulation is set so that no agent can default of its own accord.) The simulation is then run for a further 100 iterations, 2500 times, and the average evolution of each of the remaining agents' scores is recorded (given by the green line). This is then compared with the evolution of agent scores without the removal of the agent (given by the red line).

By comparison of the average score evolutions with and without agent removal one is able to see any effect the agent removal had upon the remaining agents. Rather than just studying whether an agent will default given the default of another, we are also able to see if there are any agents that benefit (in terms of wealth growth) given the failure of another.

Graphs (a) and (c) in Figure 12.21 show two different agents benefiting from the removal of another agent - this is clearly a positive default dependence. Graph (b) shows an agent that is unaffected by the agent's removal. Graph (d) shows an agent with very strong negative dependence. These results along with those of Section 12.3

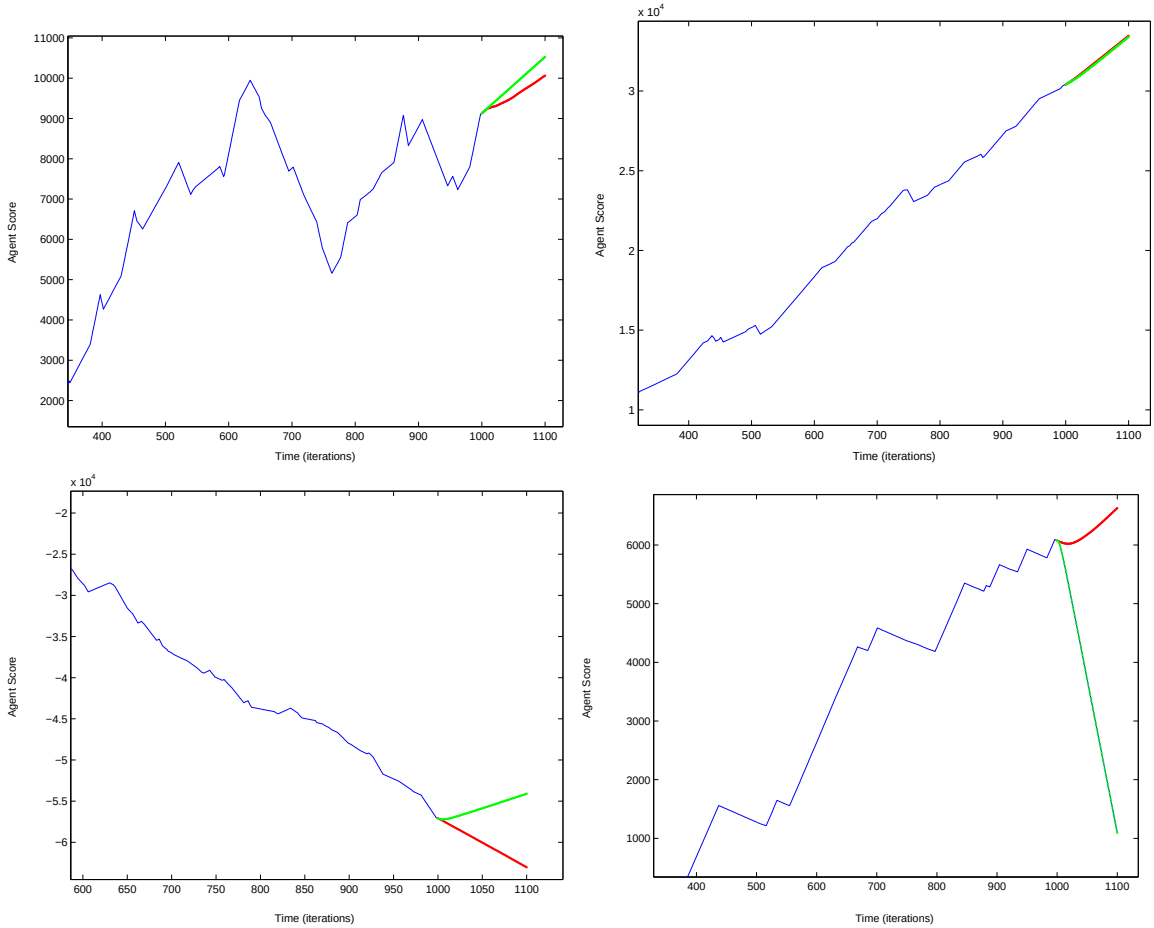


Figure 12.21: Score plots for a selection of agents. After 1000 iterations, an agent was removed from the simulation. Simulation run for a further 100 iterations, 2500 times, to compare the average evolution of agent score with (green) and without (red) agent removal with $N = 11$, $q = 9$, $s_i(0) = 100$, $z = 0.25$, $\beta = 1.0$, no default possible except for agent that is removed, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

clearly illustrate the ability of the agent network model of default to model both positive and negative default dependence.

12.12 The Dependence Matrix

The ideas in section 12.11, on measuring the dependence of agents upon the default of another, can be taken a step further by examining the dependence at a given time, t, of *all* agents to *all* others. The can be done by removing, and then replacing, each agent in turn and measuring the change to the average score evolution for each of the remaining agents.

A dependence matrix, $\underline{\underline{\Psi(t)}}$, is written as:

$$\underline{\underline{\Psi_{ij}(t)}} = \frac{\bar{s_j}(t+a|\text{default of } i) - \bar{s_j}(t+a|\text{no default of } i)}{2\beta \sum_{k=1}^{q} |R_j(k)|}, \tag{12.8}$$

where $\bar{s_j}(t+a)$ is the mean score evolution of agent j, a iterations on from time t, β is the probability that each strategy bit is assessed, q is the number of technologies or strategy bits, and R_j is the strategy score vector for agent j. Note, the score is normalised by the "size" of the agent strategy, or its potential to score points in the trading phase of the simulation, so that relative changes of wealth are compared. Therefore, element $\underline{\underline{\Psi_{ij}(t)}}$ gives the effect upon agent j if i were to suddenly default at time t. A positive value means a positive dependence, i.e., j has benefited from the default of i.

By defining some cutoff value, below which a matrix element $\underline{\underline{\Psi_{ij}(t)}}$ is set to zero, one can graphically compare the dependence matrix with the adjacency matrix $\underline{\underline{A(t)}}$.

This comparison is made using Figure 12.22. On the left of the figure is a plot of the adjacency matrix, $\underline{\underline{A(t)}}$, where t is the time that each agent is removed, i.e., these are the actual trading links between the agents at time t. The value corresponding to each node is the "size" of each agent's strategy, or its maximum potential to score in the trading phase of the simulation (the denominator in (12.8)). On the right of the figure is a graphical representation of the dependence matrix, $\underline{\underline{\Psi(t)}}$, with a cutoff value of 20.0. A negative dependence is given by a dotted line while a positive dependence by a solid line. The arrows point to indicate the effect of that agent, given default, upon others.

To aid analysis, Figure 12.23 shows the separate plots of the dependence matrix for positive and negative dependence for cutoff values of 10 and 20.

The first, and perhaps the most important result, is that while the adjacency matrix is by our definition symmetrical as it just indicates the presence of a link between

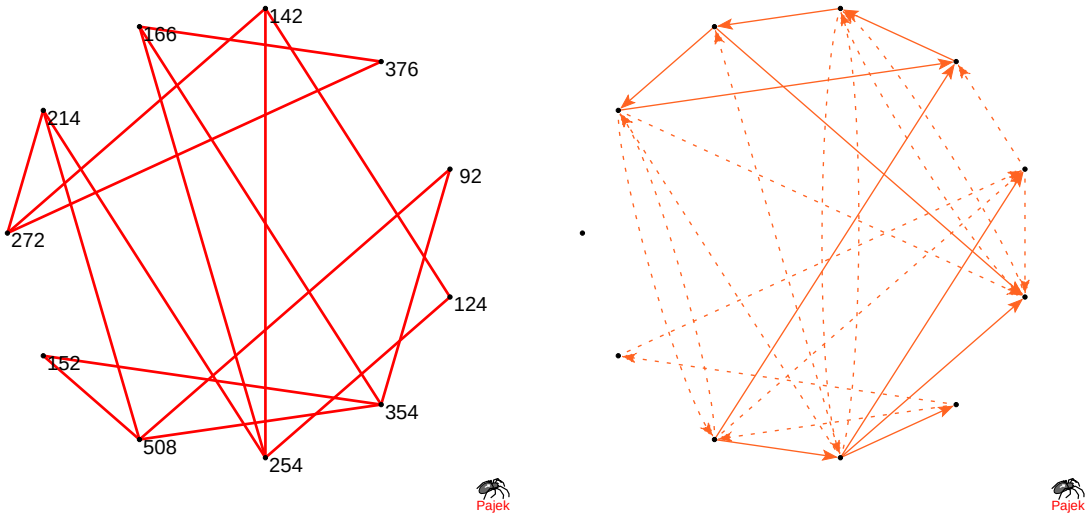


Figure 12.22: Network plot of adjacency matrix, using Pajek (left). Values next to nodes are measure of maximum potential for agent to score in trading phase of simulation (given by denominator in (12.8)). A network plot of dependence matrix with cutoff of 20.0, showing both positive and negative dependence (right). Here, $N = 11$, $q = 9$, $s_i(0) = 100$, $z = 0.25$, $\beta = 1.0$, no default possible except for agent that is removed, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

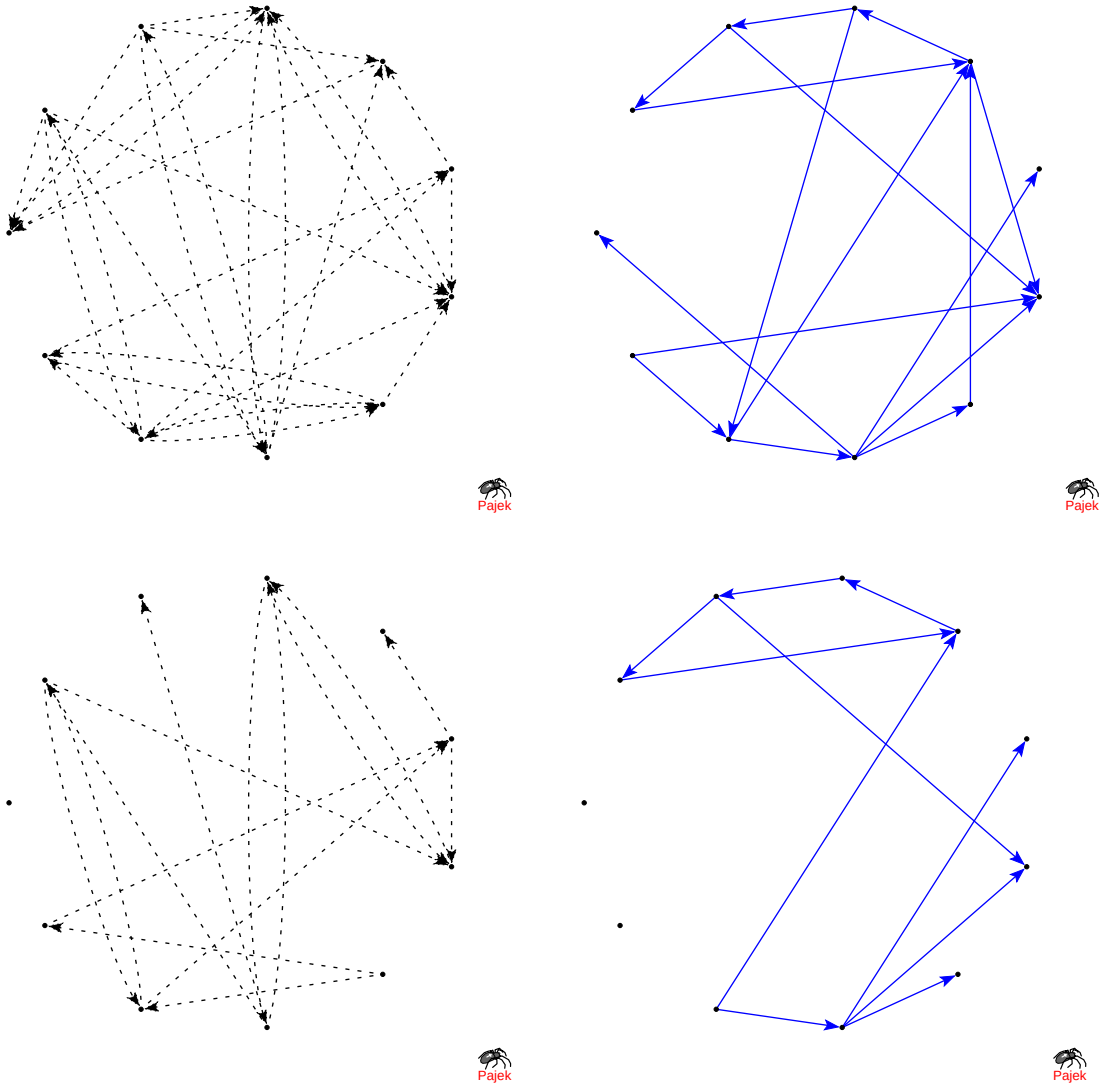


Figure 12.23: Network plots of dependence matrix, with different cutoff levels. (a) cutoff of 10.0, negative dependence. (b) cutoff of 10.0, positive dependence. (c) cutoff of 20.0, negative dependence, (d) cutoff of 20.0, positive dependence. Results from same simulations as Figure 12.22.

two agents, the real dependence matrix at this point in time is asymmetrical. There is clearly an *asymmetrical default dependence* between the agents. By comparing the plots in Figure 12.23 that relate to a cutoff value of 10.0, i.e., most of the dependencies are shown, then one can see that one agent may benefit another by defaulting whereas vice versa the dependence is negative with the other agent defaulting having a detrimental effect. This is clearly not captured by just considering the adjacency matrix, $\underline{\underline{A(t)}}$.

The adjacency matrix plot is similar to the negative dependence plot for a cutoff value of 20, shown in Figure 12.23. In fact nearly all of the links in the negative dependence plot are present in the adjacency matrix plot. This is expected as the adjacency matrix describes the direct dependence between the agents - the links they are currently dependent upon to increase their wealth and gain income, e.g., in real world terms a car dealer linked to a manufacturer. Therefore, when a neighbouring agent is removed, one would expect that on average this would have a negative effect on the agent to which it was linked.

However, in the positive dependence plot for a cutoff of 20, only one link is common with the set of links in the adjacency plot. This suggests that the agents that benefit from the default of another are not direct neighbours of the defaulting agent, unlike with the negative dependence. Therefore, this implies that some of the agents that benefit from another's default are doing so because they can form new links with neighbours of the defaulted agent which before the default may have not been optimal. Again, in context of real firms, it is comparable to two or more firms in competition. While they may not have direct business links between them, one may benefit if a competitor defaults by taking over the defaulted competitors clients.

These dependence plots clearly illustrate the difference between a direct business dependence such as a supply chain, which results in mainly negative default dependences, and indirect business dependence (such as competition - which can lead to either negative or positive dependence.

12.13 Hamming Distance as a Measure of Agent Dependence

The Hamming distance between two agent's strategy vectors, R_i is defined as the sum of the absolute difference between each of agent's strategy bits. An example is given the Table 12.1 for two 4 bit strategy vectors.

Strategy A	-2	-3	0	+6	
Strategy B	+5	-7	+4	-8	
Hamming Distance	+7	+4	+4	+14	= 29

Table 12.1: Example of calculation of Hamming distance between 2 agent strategies.

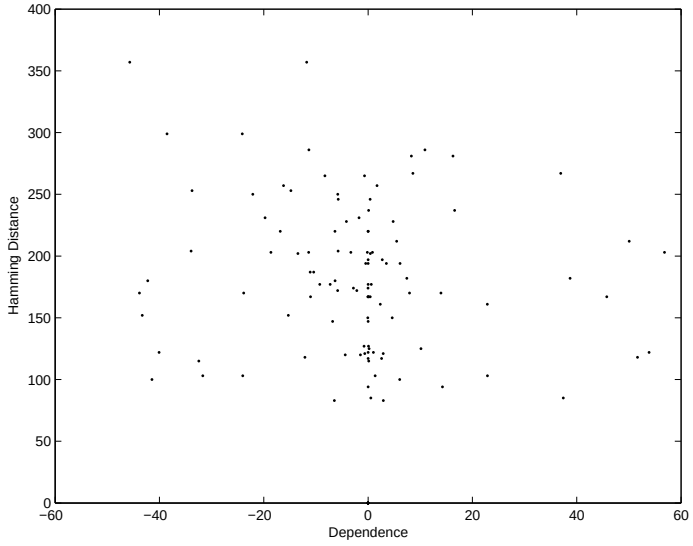


Figure 12.24: Plot of Hamming distance between two agents versus their dependence as measured by equation 12.8. For the same parameters as Figure 12.22.

By calculating the Hamming distance between all possible pairs of agents one can create a matrix of Hamming distances called the Hamming matrix, $\underline{\underline{H(t)}}$ (note the dependence on time in only relevant in the CAAM, where new agents with new strategy vectors are created throughout the simulation).

From the conclusions drawn in Section 12.12, one may expect that if one were to plot the Hamming distance between two agents against the dependence between those two agents, given by (12.8), there may be a relationship between the size and sign of dependence and the Hamming Distance. Large negative dependencies may be the result of agents with a large Hamming distance between them, as agents will try and form neighbours with strongly anti-correlated strategies which are more likely to give larger Hamming distances. Large positive dependencies, however, may be tied to agent pairs with a low Hamming distance between them. This is because, as stated above, agents with strong positive dependence are unlikely to be direct neighbours and are more likely to be in competition with each other and hence have similar strategies, with a low Hamming distance between them.

Figure 12.24 shows a plot of dependence versus Hamming distance. There is little if any visible relationship with most data points relating to approximately zero dependence. This is not altogether unexpected in one considers that the Hamming distance is only a very crude measure of the likelihood two agents are linked or in direct competition. Agents can form a whole set of neighbours and hence each strategy may be linked to a combination of other agent strategies.

Also, the above hypothesis regarding Hamming distance is very oversimplified as the removal of an agent could result in considerable reorganisation of the network with many agent benefiting or being penalised as a result of this network restructuring, rather than just those which were direct neighbours or had similar strategies to the defaulting agents. This restructuring would also help explain the results in Section 12.12 where there were many agent dependencies that bear no relation to the adjacency matrix of actual links between agents.

12.14 Network Restructuring

Figure 12.25 shows the restructuring of an agent network given the removal of an agent (agent 8). The restructured network is shown 100 iterations after the agent removal - the same time used to calculate the dependence matrices in Section 12.12. This is enough time for the other agents to suffer default.

The red square indicates the agent that has been removed and the blue triangles are agents that were neighbours of the removed agent at the point it was removed.

There has been considerable reorganisation of the network given this enforced default. For example, agent 18 is linked to 17, 8, and 7 before default. After the default it is linked to 7, 5, 15, and 1. This is the case for the majority of the agents, including those that were a few path lengths away from agent 8 such as agent 20. Significant reorganisation of the agent network after a default makes prediction of default dependencies very difficult. Even if, and is certainly almost the case, not all of the reorganisation is due to the default, the problem in predicting the real dependence between agents still remains. We have already shown that the dependencies are not only those given by the current adjacency matrix, and the network restructuring implies that we require some knowledge of the future possible adjacency matrices with their associated probabilities of occurrence as well.

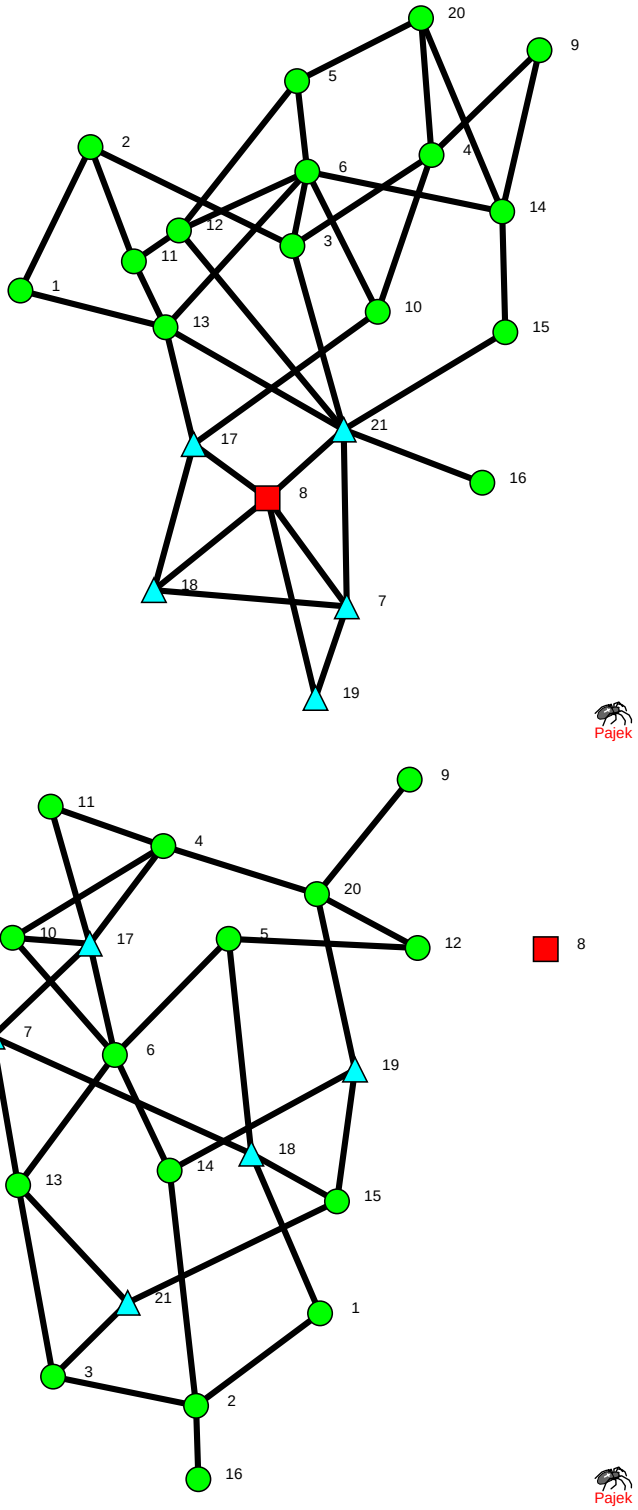


Figure 12.25: Network plots using Pajek illustrating the restructuring of the agent network given the removal of an agent with $N = 21$, $q = 9$, $z = 0.25$, $\beta = 1.0$, no default possible except for agent that is removed, $\epsilon = 0.5$, $\lambda = 1$, $\kappa_{min} = 1$, $\kappa_{max} = 50$.

Chapter 13

Conclusions and Discussion

Over the preceding chapters we have described and analysed the results of an agent-based network model of credit default. We have worked through from a simple model setup, which would incorporate the main principles and mechanisms of the final model: network formation, agent scoring and wealth, and the ability for an agent to default. The original model was modified and built upon in the hope to produce a more realistic model of market interactions, whilst still retaining some degree of simplicity in both the agents themselves and their rules of interaction. These modifications included incorporating the idea of market share - different companies will have different capacities for consumption and production of particular technologies. Also, asymmetrical scoring of a common link between two agents meant a much fairer and more realistic treatment of low capacity agents, i.e., the *man on the street* is not penalised for trading with a large company just because he does not buy all that the large company produces. An alternative default mechanism to the common threshold default, based on the individual growth of each agent's wealth, allowed for alternative interpretations of the agent score and hence a broader view of the applicability of the model to the real world. Furthermore, costs of linking and existing were included, mainly as a result of model testing, to allow patterns of default as a function of firm size to be controlled - and possibly even calibrated.

An alternative model setup was described in which there was a constant number of active agents at any point in time. This, to a very limited extent, allowed the possibility of studying the evolution of an economy in which new companies continually replace companies that default. In fact, it was shown that there is an optimal type of new agent, a rough copy of the best performing agent, if one wishes to improve the average lifetime of all agents in the network as the system evolves.

Most important was the analysis of default dependence between the agents. By studying the score evolution of the each agent, and the creation of a dependence

matrix, it was shown that both positive and negative default dependence exists within the system. Attempts to relate this dependence to the current set of active strategy vectors proved difficult. The nature of the interactions in the network meant that measures such as the application of a pseudo Hamming distance proved too simplistic to capture the complex behaviour of agent dependence and network reorganisation.

The structure of the networks was analysed graphically, and then shown not to conform to the definition of small world. Agent lifetimes were shown to have power-law tails and this provided insight into the functional form of the survival probability of agents - an invaluable commodity in the pricing of real-world default risk. A detailed study was made to discover whether clustering existed in the pattern of agent defaults. This proved inconclusive, mainly because of the very high rate of default, which is a result of the low success rate of new agents, and it was only possible to conclude that the behaviour of default was non-random. An analysis of the Hurst exponent showed there to be some persistence and hence predictability in the evolution of agent scores.

Has the model been a success and is it a realistic model of real-world firm-firm interactions and an evolving economy? For the purposes of pricing multi-name credit instruments, one would have to conclude, no. There are far too many model parameters to allow for any satisfactory model calibration - especially considering the scarcity of default data. Also, the model appears to be too dependent upon the default mechanism employed. On numerous occasions throughout the study there were shown to be significant differences in the results given a change to how agents were deemed to have defaulted. In fact it seems unfair to try and measure the success of a model such as this against such an unrealistic criterion such as its use as a pricing tool.

The model was a success if one sees it as a novel attempt to reproduce some of the behaviours characteristic of firm defaults using a simple agent scoring and linking mechanism. For many of the results, such as default clustering and agent lifetimes, it has been difficult to ascertain the validity of these as representations of real-world behaviour, because of a current lack of empirically based studies. Most importantly, however, the model clearly demonstrates asymmetrical default dependence, and not just negative default dependence, but also positive dependence, something which has been little studied and often ignored in other credit models. We believe the model sets down a platform from which either simpler or more complex variants can be developed and used to study some of the individual default "stylised facts" in detail. This fairly broad study over many of the models' properties and has shown there is

significant room for more analysis of credit using a representation of a market as a dynamic network with firms as agents and business links as edges.

Finally, natural extensions to the model include allowing a non-constant number of active agents, with the number of agents able to rise as well as fall. Then, one could study network growth and the effects of default upon the evolution of this system. Also, it would be interesting to consider a non-uniform probability distribution of agent sizes - perhaps matched to some real-world empirical data for firm size. This might provide a more realistic picture of the nature of the network structure and its evolution. Further, one might choose to use this agent network approach to model the interdependence between other entities. One example may be the trading links between nations; where agents represent countries and links represent the trading relationships, import and export of goods, etc. This model may also include the effect of the geography of countries, with agents being imposed to the constraint of a semi-rigid network structure that modelled countries physical locations.

Part III

Feedback Effects From Dynamic Hedging: An Agent Based Approach

13.1 Overview

Firstly, we study the minority game in its most basic forms, looking in detail at how the game is set up and at the results it produces, and why. This includes comparing the effect of using different reward functions for the strategy's virtual scores. Then, we look at how the basic game can be used to produce a simulation of a price process and what limitations there are to using this game as a model of a financial time series.

Secondly, we consider how the game can be adapted to produce a more satisfactory model of real world tick data, including producing the so–called stylised facts. This includes a review of the various ways in which the game can be modified and then adopting one of these methods to use in our study. The results of these modifications are analysed and any limitations to the model studied.

Finally, we adapt the minority game to incorporate a model of a trader delta hedging an option position, to see the effects that the option hedger has on the underlying minority game and the feedback that results between the option hedger and the price process, and then to try to understand why such effects occur. Then, we further adapt the game in various ways including changing the type of option being hedged and the number of times it is re-hedged to understand better the feedback within the system.

In short, by tackling the problem at a microscopic level, we want to see if the minority game can be used to give insight into the feedback effects on a market of a delta hedged option position.

Chapter 14

Introduction to an Agent Based Model of Feedback Effects in Financial Markets

14.1 The Minority Game

The minority game, as the name suggests, is an agent-based game in which a number, N, of agents make choices between one of two states. The agents choosing the state which is in the minority are then rewarded.

A very simple example of a minority game is "Zig-Zag-Zoug", a game played by children in the French-speaking regions of Switzerland. The game involves three children putting their right foot into the centre of a circle. The children say together "Zig-Zag-Zoug", and on Zoug the children must either remove their foot from the circle or leave it there. The winner is the child who is the minority.

The minority game is an abstraction of a problem stated by W.B.Arthur called the El Farol bar problem [52]. In this problem 100 people have to decide whether to go to a bar called El Farol on Thursday night to listen to some Irish music. As space in the bar is limited the evening would only be enjoyable if fewer than 60 people go to the bar. However, if 60 or more decide to turn up then the evening would no longer be enjoyable and the people would have been better off staying away. There is no way that the people can work out exactly how many others will attend, so they must base their choice on what they think the other people will do. They must do this by forming an expectation of how busy the bar will be. This is a crucial point: the people cannot use deductive reasoning, and must move into a regime of inductive reasoning. But, if all the people share the same expectation then they will all be wrong. If all believe few will go, then all will attend, and the bar would be too busy

to enjoy. In the same way as the children playing "Zig-Zag-Zoug" must make a choice based on an expectation different from the others, so must the people attending the bar. If everyone does the same thing then no one is a winner.

These two ideas, agents using inductive reasoning, and agents being forced to form different expectations, are the main mechanisms behind the minority game and are what make it such a rich model to study.

D.Challet and Y.C.Zhang first looked at the form of the minority game (and in fact coined the term minority game) that is the basis of this study [53]. We deal with the details of the model later in the report but will point out the main features now. An odd number, N, of agents are given a finite number of strategies which they use to choose between two states. The strategies are a way for the agents to form an expectation of the state that will be the next minority outcome (i.e., the state which is chosen by the minority of agents). The various strategies available to each agent have some quantitative measure of success (the precise form of this measure varies) which is increased if the strategy successfully predicted the minority state in the previous iteration of the game and not increased (or possibly decreased) otherwise. The agents choose the currently most successful strategy (i.e., the strategy currently with the highest measure of success) to predict the outcome of the next iteration of the game. Over time the strategies an agent owns begin to distinguish themselves from one another by their success at predicting the minority outcome. As mentioned above, if all agents had access to and used the same strategy then this would no longer be successful and other strategies would become optimal. The agents are making decisions based only on a common knowledge of the past minority outcomes. They know nothing about the other individual agents or their strategies, just the result of the collective.

This fairly simple model has led to a large and rapidly growing body of study on the subject of agent games based around a minority reward mechanism [54].

Many of the early studies consists of qualitative and quantitative explanations for observed properties of minority games including a phase transition that occurs in the variance of the number of agents choosing a particular state at each iteration. As the memory (which is the number of past minority outcomes over which each strategy is allowed to look) is altered, the variance moves, first from a maximum value for low memory, then to a minimum value, then to a value that is the same as it would be if the agents were choosing each outcome at random. An intuitive approach to this problem is the crowd-anticrowd theory [55][56]. This proposes grouping together

the agents that are using similar strategies and modelling the behaviour of the whole population of agents in terms of the size of these groups.

Other approaches to analysing the properties of the minority game include using techniques from magnetic theory [66] or writing stochastic differential equations and describing the game in the continuous time limit [57].

Because of its simplicity, the minority game lends itself to a large number of adaptations. These include incorporating Darwinist selection into the game by periodically replacing the least successful agents with clones of the best agents [53]. Also, the game can be set up with agents that have different memory lengths [58]. This means that some agents will have strategies that look further into the past than others, providing them with more information to work with. We consider this last point further when adapting a minority game to model a financial market.

14.2 The Minority Game as a Market Model

The claim that the minority game can be used as the basis for a model of a financial market may seem bold. However, a lot of the studies into the minority game centre around this very idea with some interesting results (again the reader is referred to the minority game's website to review all of these studies [54]). The idea of applying the minority game to finance is not to exactly replicate a financial market, but rather to use it as a simple model with a mechanism that is comparable, at least over short time scales, to those that drive real world asset prices. One of the major assumptions behind the Black-Sholes model is that market dynamics (or the price of an asset or financial instrument) can be, at the most fundamental level, modelled as a random walk [60]. For shares, this random walk takes the form of geometric Brownian motion, the details of which we discuss later in the report. The minority game can be used to produce a price process in the same way as geometric Brownian Motion, but whereas just assuming a deterministic drift with a random component as a model of the evolution of the price (as is the case the geometric Brownian motion), by using the minority game, properties from real world markets can be incorporated. Further into our study we show how certain stylised market facts can be replicated by extending the basic minority game, but first it is instructive to look in a little more detail at the assumption that the minority game can be thought of as a market model.

If one thinks of the agents in the minority game as traders - somewhat naive though this may be - then if all the traders enter the market at the same time, some wanting to buy, some sell, whichever group is in the minority is in the best position.

For example, if we are selling something, we would prefer for everyone else to be buying, and if we want to buy something, we would benefit if everyone else were wanting to sell. The difference between the number buying to the number selling can then be used to vary a price process, via the law of supply and demand.

These comparisons to the market can be taken further. If one considers a market as a place for reallocating goods, then as no arbitrage (riskless profit) means that no gain can be made by pure trading alone, transaction costs make it a game where *on average* you are likely to loose, i.e., a minority game. This is of course, very simplified. It doesn't take account of speculative players who buy an asset with the intention of holding onto it until the price has risen and then going back to the market to sell it at a profit. This would be a majority game where an agent would want most other agents to act as they are and force the asset price in a particular direction – so called trend followers.

It is, however, reasonable to assume that traders form an expectation of a market (use their strategies) before going into that market and acting on those expectations – bounded rationality [1]. Then, over time these expectations are modified (agents choose different strategies) if the traders are not performing well compared to the rest of the market. The traders are using induction rather than deduction. From a market mechanism such as this, a version of the minority game can be derived [59]. This demonstrates the parallels between the actions of real world traders and minority game agents.

Hence, though simple, the minority game shares a number of significant features with short term trading in real world markets.

14.3 Geometric Brownian Motion

The idea of Brownian motion as a model of asset price fluctuations was first introduced in 1900 by Louis Bachelier in his work 'Theorie de la speculation'. (A good description of Brownian motion is given in [60].) Two significant flaws in this model are that the asset price can become negative and that interest rates are zero [2].

Case Sprenkle, James Boness and Paul Samuelson, in the 1960's, improved upon Brownian motion by introducing geometric (or lognormal) Brownian motion. The

[1] Rationality refers to individuals behaving in such a way as to maximize their self interest. When constraints such as assuming the individual has only incomplete information are incorporated, then this becomes bounded rationality.

[2] The only asset (not derivative) we can think of that might have a negative price is toxic waste.

formula for this is:

$$dS = \mu S dt + \sigma S dX \tag{14.1}$$

where μ is the drift of the asset, σ is the volatility of the asset, and dX is the change is a random walk described by Brownian motion, $X(t)$. As is clear from the formula, if the asset price starts off positive it can never go negative. The drift μ adds trending to the price process. An important point is that geometric Brownian motion is continuous. This is in contrast to the discrete nature of the minority game.

Geometric Brownian motion was then used by Fischer Black, Robert Merton and Myron Scholes in 1969 to develop the Black-Scholes option pricing framework [62][63].

14.4 Black-Scholes, Delta, and Gamma

A European call or put option is the right, but not the obligation, to buy or sell a particular asset for an agreed amount at a specified time in the future. A call option is the right to buy the underlying asset, a put option is the right to sell the underlying asset. The strike price is the value at which the underlying asset can be bought (call) or sold (put)[61].

The Black-Scholes partial differential equation for the value of an option contract is:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{14.2}$$

where V is the value of the option, S is the value of the underlying asset, t is the time to the maturity of the contract, r is the risk-free interest rate, and σ is the volatility of the underlying. There are many assumptions behind the Black-Scholes formula [60]. Two assumptions particularly relevant to our study are; the underlying follows a lognormal random walk, and that delta hedging can be done continuously. The first assumption given above is clearly not the case for our study as we use the minority game to generate the price series for the underlying asset. We come back to the second assumption later in the report.

The value for a vanilla call option contract on an underlying paying no dividends is:

$$SN(d_1) - Ee^{-r(T-t)}N(d_2) \tag{14.3}$$

where

$$d_1 = \frac{\log(S/E) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \tag{14.4}$$

and

$$d_2 = \frac{\log(S/E) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \tag{14.5}$$

and E is the option strike.

The delta is the sensitivity of the option to changes in the value of the underlying. As can be seen from the formula below, it is the rate of change of the value of the option with respect to the value of the underlying asset:

$$\Delta = \frac{\partial V}{\partial S} \tag{14.6}$$

Delta hedging is an example of dynamic hedging. If one holds one option contract, then to be delta hedged, one must also be short a quantity Δ of the underlying. This is a delta neutral position - any change in the value of the option will be accompanied by an equal and opposite change in the value of the short quantity Δ of the underlying, to a first order approximation. The formula for the delta of a vanilla call is:

$$N(d_1) \tag{14.7}$$

where d_1 is given by equation 14.4. As can be seen, the value of delta depends on both S and t. Therefore the value of the delta hedge must continuously be changed as the contract moves towards maturity and the value of the underlying changes. This is our second assumption from above. In real world markets this is impossible as the physical action of trading means that it must be done in discrete time. Also, the transaction costs involved in buying and selling the underlying can make it prohibitively expensive to rehedge too often. As stated in the discussion on geometric Brownian motion, the minority game model itself operates in discrete time, and so it would be impossible to rehedge continuously. The assumption of applying Black-Scholes to real world markets is the same as applying it to a minority game market where each iteration of the game is one time unit.

The gamma of an option is the second derivative of the option value with respect to the underlying asset:

$$\Gamma = \frac{\partial^2 V}{\partial S^2} \tag{14.8}$$

Gamma is the sensitivity of the delta to changes in the value of the underlying. It is therefore a measure of how often, or by how much, a position must be re-hedged to remain delta neutral.

14.5 Feedback in financial markets

In our report we look at the effect of delta hedging an option on the price of the underlying asset, i.e., what is the effect on the price of an asset if we take positions in that asset to delta hedge an option on that asset? This feedback, implied by the Black-Scholes trading strategy is completely ignored by the classical Black-Scholes model.

Some work has already been done to study the effects of feedback in financial markets. Brief details of one approach that uses the Black-Scholes framework are given later in the report. Other approaches can be found by referring to [71].

The minority game lends itself as a good platform on which to study feedback effects, as the price process it creates is the result of the interaction between many *independent* agents. This allows the properties of an agent to be adapted so that its response to changes within the system, such as price or history, can be modelled on a trader with a particular mandate, such as to hedge an option. The actions of the modified agent will then have an impact on the price or whatever output the agent is responding to. Hence, the system easily incorporates within it the interaction between a market and someone acting on the changes in that market, whilst also allowing a feedback mechanism to operate between the actions of the modified agent and its inputs. This change to the price process is happening at the most fundamental level. Rather than set up a process such as geometric Brownian motion and then modify the mathematical models that derive from this process with terms to include any feedback mechanism, the minority game will allow this mechanism to be introduced at beginning, as the price process is constructed.

Chapter 15

The Vanilla Minority Game

In this chapter we look in detail at the most basic of the minority games. We show how the game is set up and then used to create a price process. We look at empirical studies into the minority game and theoretical descriptions of these results.

15.1 Setup of the Vanilla Minority Game

The vanilla minority game[1] consists of a population of N (odd) agents. Each agent has, the same, finite number of strategies S. The agents use their strategies to decide, at each iteration of the game, whether to be in one of two states, 0 or 1. If the agent is in the state that is in the minority then it is rewarded with a point.

The only information that agents have access to is the history of minority outcomes, i.e., the binary string of which state was chosen by the minority of agents at each iteration. For example a history string might be ...011000011100101, with the most recent history on the right. The memory, m, of the agent is the number of bits of the history string it can use to make the next decision. An agent with a memory of 3 can look at the last 3 bits in the history string, i.e., in the example above this would be 101. The agents' strategies are bit strings of length 2^m. For each possible bit string of history length m they have a choice of state, 0 or 1 as shown in Table 15.1. For the strategy shown in this table, if the history was 101, then the strategy would tell the agent to predict that state 0 will be in the minority. The strategy space is the 2^{2^m} different possible strategies that an agent could be given at the beginning of the game. As the strategies are allocated to an agent at random it is possible that an agent could hold two identical strategies.

[1] In our report we refer to what we believe to be the class of most basic minority games as vanilla minority games (VMG) in the same vein as the labelling of option contracts as vanilla or exotic[60].

history	prediction
000	1
001	0
010	1
011	1
100	1
101	0
110	1
111	0

Table 15.1: The predictions of one strategy for all possible 3 bit history strings.

Each strategy that an agent holds will have a virtual score which is incremented by a payoff based on whether the strategy made a correct prediction. This happens whether or not the strategy is actually in play. Agents decide which strategy to use by whichever has the highest virtual score. If two or more strategies have the same virtual score then the result is decided at random using a uniform distribution.

15.2 Generating a Price Series

There are numerous methods for generating a price series from the minority game. One is to increase a price by one unit if the minority outcome is 1, and to decrease by one unit if the minority outcome is 0. Another method – the one we have chosen – is to consider, at each iteration, the population of agents choosing each of the states. This allows the price to move in accordance with the weight of the number of agents wanting to buy compared to the number wanting to sell. If nearly all of the agents choose state 1, they go into the market wanting to buy. Hence very few choose state 0 and there are not many sellers. In this simulation you would expect the price to rise more than if the number of people buying and selling were almost the same.

More formally, consider the history bit string for the minority state as the integer variable μ, which can lie between 0 and $2^m - 1$, which we will call the P possible histories. Let $a^{\mu}_{s,i}(t)$ denote the prediction at time t, of agent i (where $i = 1, ..., N$), using strategy s (where $s = 1, ..., S$), for a given history μ. For our setup, if agent i predicts a sell, or 0, then $a_i(t) = -1$, and if agent i predicts a buy, or 1, then $a_i(t) = +1$ [2] . Labelling $a_i(t) = -1$ for a sell prediction rather than 0 makes the algebra simpler.

[2]We let $a_i(t)$ represent $a^{\mu}_{s,i}(t)$ when we are talking in general and not considering a specific history or strategy.

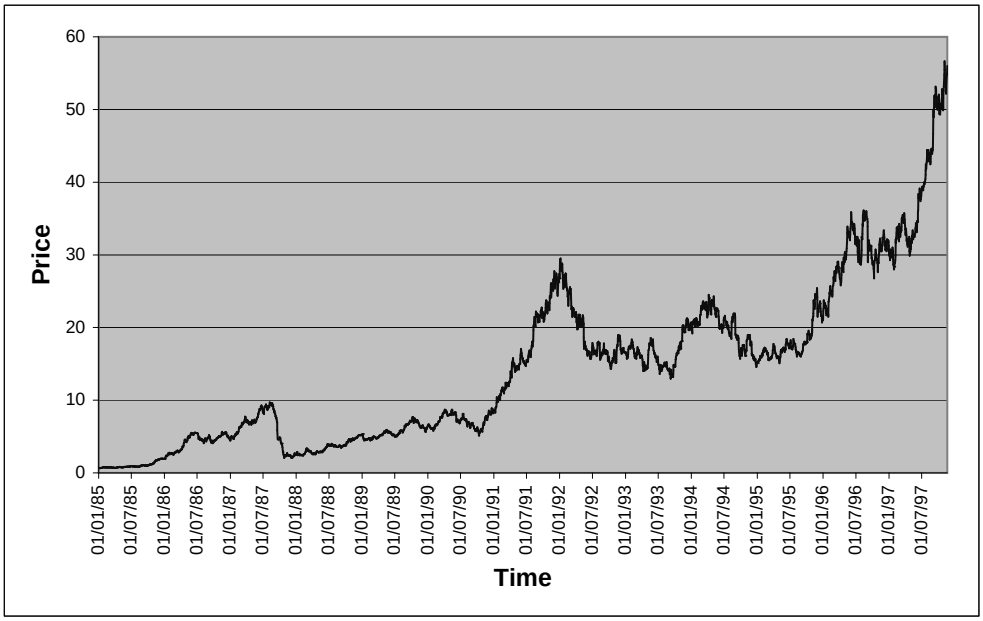


Figure 15.1: Daily closing price versus time, for stock ARV Assisted Living Inc. Data from Yahoo Finance.

Let $A(t) = \sum_i a_i(t)$, the sum of all the agents predictions. This is indicative of the demand for an asset. If $A(t)$ is large and positive, then most of the agents want to buy, and you would expect the price of the asset to rise significantly. If it is large and negative, then most of the agents want to sell, and you would expect the price of the asset to drop significantly. However, if $A(t)$ is close to zero, then demand is matched by supply, and you would expect the price to remain approximately where it is.

As our goal is to model the price series for a real world market or asset then we must take into consideration some attributes of real prices. In our study we consider the asset to be a share or equity in a company, and the market to be a stock market such as the FTSE. Firstly, the price of a share will never go negative. Secondly, the variations in the price of the share increase as the share price increases, see figure 15.1.

Therefore, to take account of these features, the formula for our price at time t, $p(t)$, is taken to be:

$$p(t+1) = p(t)e^{\alpha A(t)} \tag{15.1}$$

where α is some positive constant that relates the size of the excess demand or supply, $A(t)$, to the size of the price change.

15.3 Comparing Virtual Score Reward Functions

One simple reward mechanism is to increment the virtual score of a strategy by one point if it makes a correct prediction. The virtual scores of strategies that make an incorrect prediction remain unchanged.

Alternatively, the virtual scores, like the price process, $p(t)$, are updated based on the value $A(t)$, i.e., the size of the reward is based on the number of agents making the same choice as to what the minority outcome is. If the virtual score at time t is represented as $U_i(t)$, then the scores are updated according to:

$$U_i(t+1) = U_i(t) - a_i(t)A(t) \tag{15.2}$$

where a_i is the prediction -1 or 1 of agent i. As can be seen from the formula, the virtual scores can fall as well as rise.

The choice of reward is studied later in the report and most suitable reward to use in a market model is assessed. In both the methods above, the strategies are rewarded for making a correct prediction and penalised for an incorrect prediction. It may seem at first as if changing the payoff will not have a significant effect on the game. Later I show that in fact it does.

15.4 Properties of the Vanilla Minority Game

We are going to present some of the main properties of the vanilla minority game that have been widely studied. These are useful in gaining an understanding of the game and an intuition for its behaviour.

The value $\langle A \rangle$ is the average over time of the value $A(t)$, the sum of all the agents' predictions, and over all histories, μ. Due to the symmetry of the system – the 2 states are identical and the selection of agent strategies is random – this value is zero, as one might expect (i.e., $\langle A \rangle = 0$).

However, the value $\langle A^\mu \rangle$, which is the average over time of $A(t)$ for a given history μ, has been shown to be non-zero for some values of μ, $\langle A^\mu \rangle \neq 0$ [64, 66]. This means that for some histories that may occur, the resulting $A(t)$ may be more likely to take a positive (or negative) value.

The non-zero value of $\langle A^\mu \rangle$ has been referred to as exploitable information, with $H = \frac{1}{P}\sum_\mu \langle A^\mu \rangle^2$, where P denotes the $(2^m - 1)$ possible histories, being a measure of the exploitable information content of the system. The idea of exploitable information means that if an agent were to join the game after it had been played for some time,

then there is a "best" strategy that will ensure that overall it will be in the minority most of the time. If $\langle A^\mu \rangle$ is non-zero, therefore for some values of history, μ, the game will on average have a bias towards the number of agents choosing one outcome over the other outcome every time this history occurs. So, a strategy exists that will, on average, successfully predict the minority outcome more often than not, or at least benefit from the increased weight of agents choosing one outcome over the other, effectively exploiting the other agents and this information they have created. When $H = 0$, there is no information to exploit and hence all $\langle A^\mu \rangle = 0$. Note that even though $\langle A \rangle = 0$, this does not mean that all $\langle A^\mu \rangle = 0$. This explains why the value H looks at the square of $\langle A^\mu \rangle$.

It has also been shown that there is a phase transition in the minority game when it moves from what is called a symmetric phase to an asymmetric phase as the size of the strategy space increases in relation to the number of agents. The symmetric phase is where $H = 0$, i.e., there is no information in the system to exploit. In the asymmetric phase $H > 0$, and there is information for an external agent to exploit [66]. In the symmetric phase the game exhibits an anti-persistence - a particular history that initially meant an outcome of one minority state will be followed by the opposite outcome when that history occurs again. In this phase the game will cycle through a set pattern of histories until randomness (introduced by the agents randomly choosing between several strategies with equal highest virtual scores), maybe forces it into a different cycle of histories. In the asymmetric phase however, the game's outcomes display persistence.

In addition to the exploitable information, another empirical property of interest in the minority game is the volatility of the system, $\sigma^2 = \langle A^2 \rangle$ [66]. This is a measure of the collective efficiency of the agents playing the game. If σ^2 is large then, on average, the difference in the size of the groups of agents choosing each state, $A(t)$, is much larger than is necessary, if the hope is that as many agents as possible are rewarded. The most efficient outcome at each iteration is that only $\frac{N+1}{2}$ agents are left in the majority group. A small value of σ^2 means good coordination between the agents. It is as if they are working together to ensure that even though only a minority are rewarded at each iteration, that this minority is as large as possible.

As with the value of H, the volatility varies with the number of agents playing the game, N, and the size of the strategy space (which is determined by the memory size m). As can be seen in figure 15.2, as the value of the memory is increased for a fixed N, the volatility first moves from a maximum value at $m = 2$, down to a minimum, and then back up towards a value equal to the volatility if the agents were making

their decisions at random based on coin tosses. Again, the phase transition between the symmetric phase and asymmetric phase is evident with a reduction in volatility with m for the symmetric phase and an increase in volatility with m occurring in the asymmetric phase. The value for the memory at which the volatility is a minimum, i.e., the value of the boundary between the symmetric and asymmetric phases is at $m_c \sim \ln N$ [65].

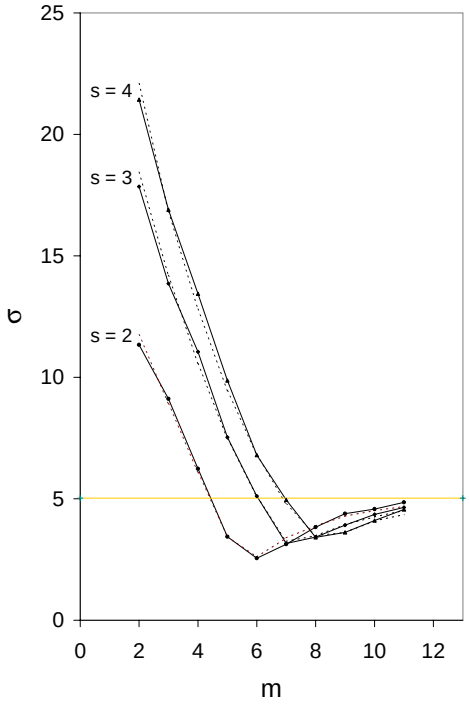


Figure 15.2: Standard Deviation σ for the minority game as a function of memory size for $s = 2, 3, 4$ strategies per agent and $N = 101$ agents. (For reduced strategy space) Graph taken from [56].

This feature of the volatility can be described quantitatively, either, as an exact solution in the limit of infinitely many agents, using a spin formulation from magnetic theory [66], or by looking at the agents as crowds with correlated strategies [55, 56].

The crowd-anticrowd theory [55, 56] of the minority game also provides an intuitive qualitative explanation for the change in volatility with agent memory m. Consider a fixed number of agents N. Let $i^\star$ be the best strategy, out of all strategies available, for a particular moment in the game, i.e., the strategy with the highest virtual score. There will also exist strategies that are very much like $i^\star$, i.e., they only differ from $i^\star$ by a small number of bits relative to the overall size of the strategy. These should also have high virtual scores.

If the agents' memory, m, is small, the corresponding strategy space is also small and the chance of any agent having strategy $i^\star$ or one very much like it is high. Hence, the chance of an agent having *only* poorly performing strategies to choose from is low. Therefore, as most of the agents have similar best performing strategies they will act as one large group doing the same thing at each iteration of the game. As one would expect this would lead to a large absolute value of $A(t)$ and hence a large value for σ^2.

For large m, the strategy space is very large, and with high probability, the strategies are spread over the agents such that there is a very small chance that any agents will share the same strategies. Whatever best strategy $i^\star$ that exists is probably either not used or is used by just one agent. Therefore, the agents act independently and there is a low chance of them forming a crowd of agents all acting in a similar manner. Hence, because of the independent action of the agents the value of σ^2 approaches the same value as if the agents were randomly choosing a state based on the flip of a coin.

If, however, we are at an intermediate value of m, with a relatively large strategy space, then there is a good chance that some agents' strategies will all be low scoring, i.e., whose prediction for each possible history string, μ, is nearly the opposite to that of $i^\star$. Agents with these anti-correlated strategies, $\bar{i}^\star$, form what are called anticrowds and act in an opposite manner to the crowds of agents with the highest scoring strategies. The interaction between these two crowds reduces the value of $A(t)$ due to cancelling out, creating the minimum in the volatility.

If the value of an individual agent's variance, that has to choose between state 1 or 0, is $\sigma_i^2 = \frac{1}{4}$, then if one considers each agent to be independent, the total variance of all N agents acting together is just the sum of the individual variances [3]. Hence, for large m, the variance of the system is approximately equal to $\frac{N}{4}$. When m is small and the agents are acting as one large crowd and not independently, the variance of the group of size N is approximately $\frac{N^2}{4}$ (consider one large crowd choosing between N lots of state 1 or 0). For intermediate m, in the extreme case when crowds and anticrowds are of similar size, the value of σ^2 is of the order 0, as all the actions of the crowd are cancelled out by the actions of the anticrowd.

Therefore, although approximate, by studying the minority game from the perspective of crowds and anticrowds, real insight into the behaviour of the system can be gained.

[3] Assume agent i can chooses between state 0 or 1 with equal probability. The mean value of the prediction is $\frac{1}{2}$ and hence the variance $\sigma_i^2 = \frac{1}{2}\Big((1-\frac{1}{2})^2 + (0-\frac{1}{2})^2\Big) = \frac{1}{4}$.

15.5 Results for the Vanilla Minority Game

The results we present here for the vanilla minority game are for two different virtual score rewards. The first set of results is for the reward of +1 point for a correct prediction and this will be referred to as the *step reward*. The second set of results is for the reward given in equation 15.2 and will be referred to as the *linear reward*.

step reward (only for a correct prediction):

$$U_i(t+1) = U_i(t) + 1 \tag{15.3}$$

linear reward:

$$U_i(t+1) = U_i(t) - a_i(t)A(t) \tag{15.4}$$

Figure 15.3 shows the change in the price generated by the vanilla minority game with a step reward over a large number of iterations.

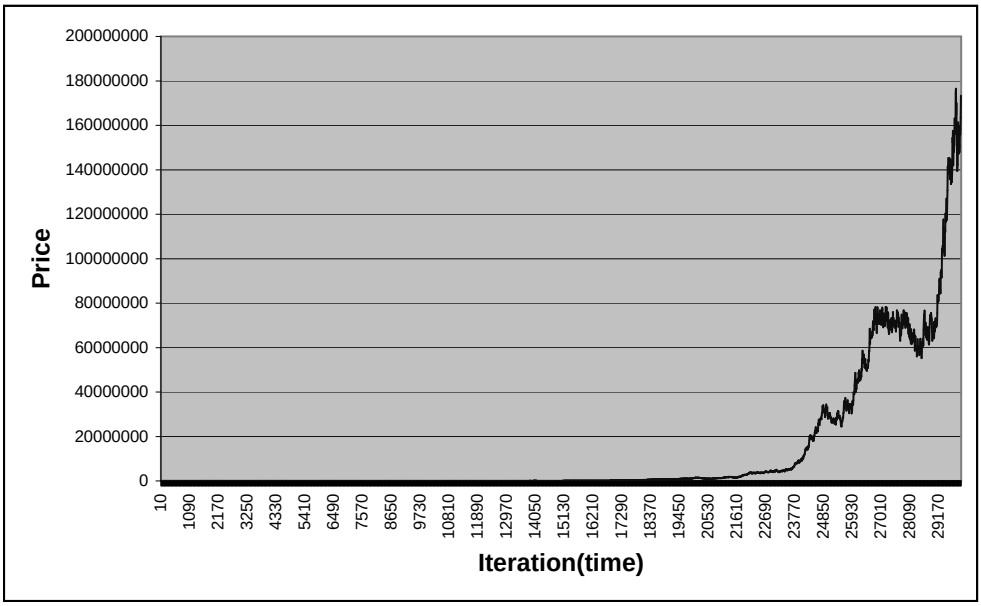


Figure 15.3: Change in price with time over 30,000 iterations for vanilla minority game with step reward, $m = 3$, $s = 2$, $N = 1001$.

A detail of figure 15.3 is given in figure 15.4. Over the smaller time scale the results look similar to those that may be produced by geometric Brownian motion. However, closer inspection shows a clear repetition of a pattern of price changes which

you would not expect to find in geometric Brownian motion. This, as stated earlier, is due to the minority game cycling though a set pattern of histories until randomness introduced by two or more strategies having the same virtual scores changes this pattern.

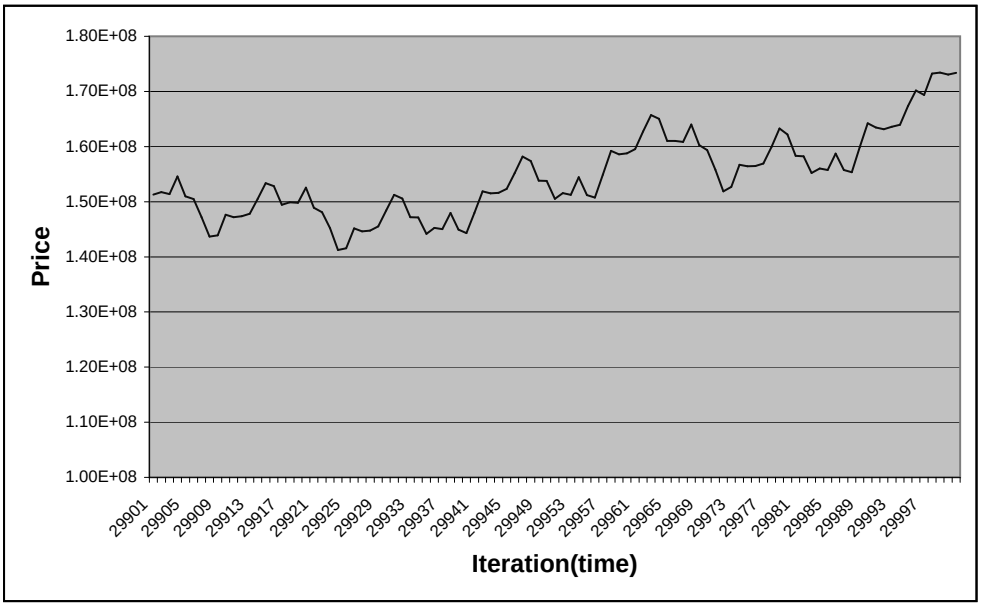


Figure 15.4: Detail for figure 15.3, $m = 3$, $s = 2$, $N = 1001$.

Figure 15.5 shows a histogram of the returns generated from a vanilla minority game price series with the step reward. The "head and shoulders" appearance of the triple peaked distribution is very different from the Normal distribution produced by Brownian motion.

An explanation for this shape distribution is provided by M.A.R. de Cara [72]. Consider as the efficient regime the part of the symmetric phase where the volatility, $\sigma^2 = \langle A^2 \rangle$, is greater than if the agents were deciding at random. In this regime almost all possible strategies are in possession of the agents. Therefore, the *current* best performing strategy $i^\star$ is owned by a lot of the agents who will all act together. Up to now this is the same as the crowd anti-crowd qualitative description for low m. Now, consider the anti-correlated strategy $\bar{i}^\star$. This will also be in possession of a lot of the agents although not necessarily all those that own $i^\star$. Assume that the number of agents that own $i^\star$ is approximately equal to the number that own $\bar{i}^\star$, and denote this as n_{correl}. Now denote as n_{random} the number of agents that don't own the current best strategy (whether it happens to be $i^\star$ or $\bar{i}^\star$ at that point in the game), as their choices can be taken as random.

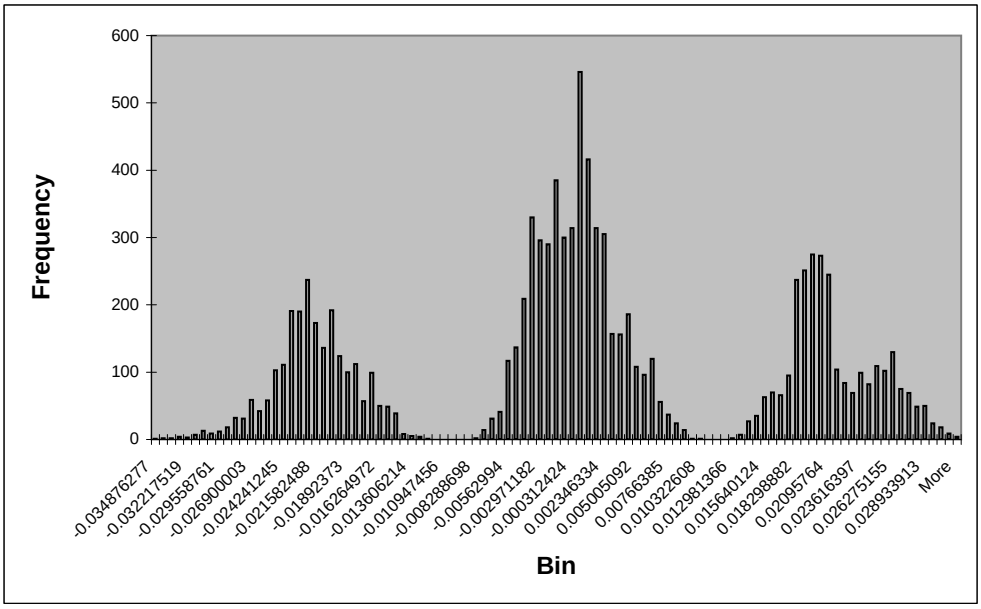


Figure 15.5: Histogram of returns for vanilla minority game with a step reward, $m = 3$, $s = 2$, $N = 1001$.

All the time $i^\star$ is the best performing strategy there will be two groups, one choosing the same outcome as predicted by $i^\star$ which will be of size $n_{correl} + n_{random}/2$ and a group of size $n_{random}/2$ doing the opposite. Hence, the size of the group using the current best strategy is added to by half of the random group. The combination of n_{correl} and $n_{random}/2$ produces the side peak or the "shoulders".

This outcome would give no virtual points to $i^\star$ but would give points to $\bar{i}^\star$ until the virtual scores of both are approximately the same. Then, the two groups choosing either prediction will be approximately $N/2$ - agents with only $i^\star$ will use that, and agents with only $\bar{i}^\star$ will use that, while agents with both will choose randomly as will agents with neither. This creates the middle peak.

The system will then move back to either $i^\star$ or $\bar{i}^\star$ being the best performing strategy. Overall, the game fluctuates between two unequal group sizes (side peaks) and two equal group sizes (middle peak).

The formation of the side peaks can also explain the large value of the volatility for lower memory. From figure 15.5 it is clear that the presence of the peaks greatly contributes to the variance of the distribution. For large values of memory the game moves out of the efficient regime into an inefficient regime and the side peaks disappear. The very much larger strategy space means one can no longer assume that large groups of agents hold the same strategy and act together, but rather that they act randomly. Again, this is similar to the crowd anti-crowd theory for high m.

Figure 15.6 shows the change in price generated by the vanilla minority game with a linear reward. Figure 15.7 is a detail taken from figure 15.6. Again, as with the step

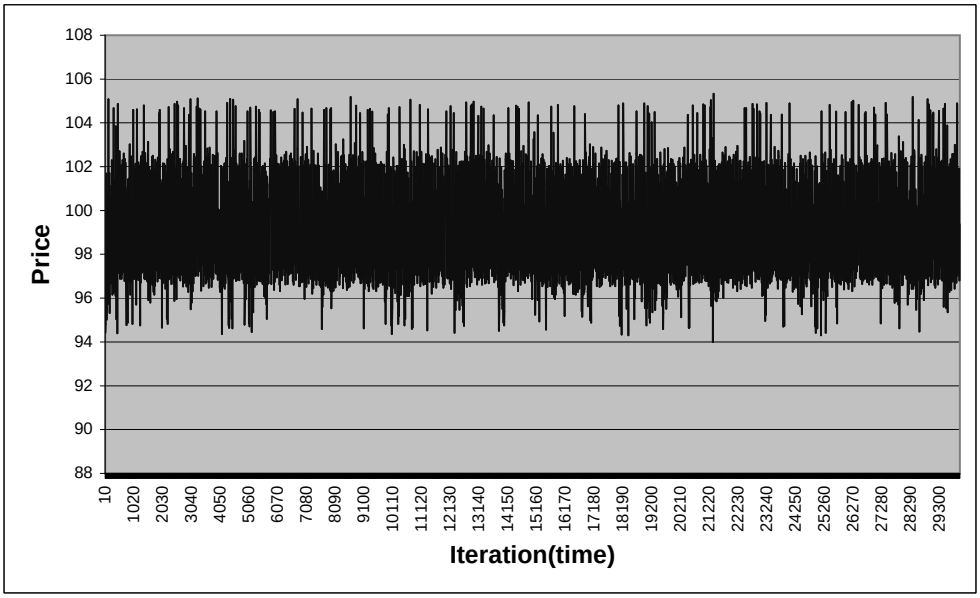


Figure 15.6: Change in price with time over 30,000 iterations for vanilla minority game with a linear reward, $m = 3$, $s = 2$, $N = 1001$.

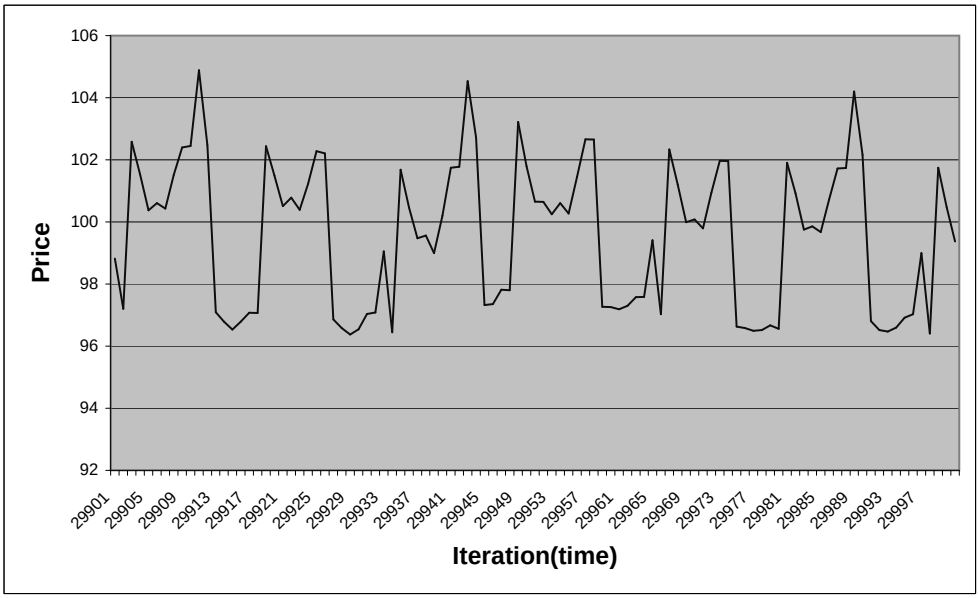


Figure 15.7: Detail for figure 15.6, $m = 3$, $s = 2$, $N = 1001$.

reward, there is a strong pattern in the price series. There are only small fluctuations to this pattern which are due to tied virtual scores. The pattern itself will only change

when the combined contributions of all agents with tied virtual scores are enough to change the minority outcome itself. A better understanding of these patterns can be gained by analysis of a de Bruijn graph which shows diagrammatically the possible history outcomes of the game and the cyclical patterns between the histories that can occur [67].

A histogram of the returns for the linear reward vanilla minority game is shown in figure 15.8. Again there is more that one central peak - in fact now there are at least

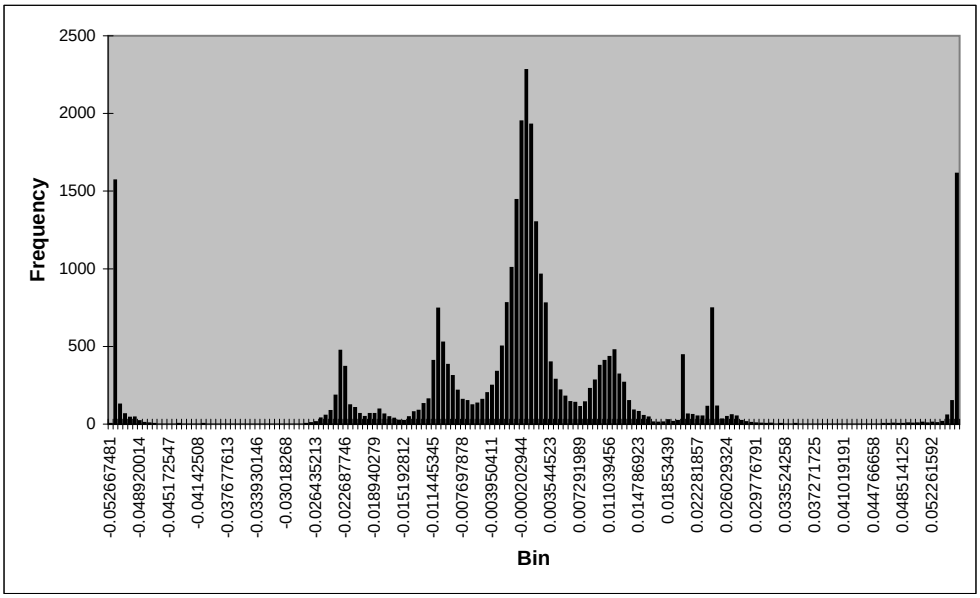


Figure 15.8: Histogram of returns for vanilla minority game with a linear reward, $m = 3$, $s = 2$, $N = 1001$.

five clear peaks. We think the general argument by de Cara for additional peaks still holds for the linear reward. The assumption of just one dominant anti-correlated pair of strategies competing with each other may need to be lifted to explain the extra peaks not seen for the step reward. (This is looked at in his paper as is the effect of changing the reward [72].) Also from the histogram one can see that the central peak is much larger than the side peaks in comparison to the histogram for the step reward. This may be due to the linear reward reducing the score of poorly performing strategies as well as rewarding the successful strategies, whereas the step reward only rewards the successful strategies. Therefore, the linear game will move more quickly to a position where strategy scores are the same and less time when they are different. This would mean a greater contribution to the middle peak as opposed to the outer peaks.

15.6 Limitations of the Vanilla Minority Game as a Market Model

As was seen, both games show strong repetition in their price series. One needs to consider if this attribute is a fairer model of price dynamics than that of Brownian motion. Just because real world price series are considered not to be totally random does not mean that the patterns produced by the vanilla minority game are better models.

Also, the multiple peaks in the return histograms, which is a product of the game dynamics, is a limitation of using the vanilla minority game as this effect (as will be shown later) is not representative of true asset returns.

Chapter 16

The Stylised Minority Game

From the study of the vanilla minority game it is clear that it needs adapting before it can be considered as a model of the financial markets. In this chapter we show how the game can be adapted to produce some of a market's stylised facts with the creation of a stylised minority game.[1]

16.1 Stylised Facts

Financial data exhibits characteristic non-trivial properties that have been labelled stylised facts. There is no exact definition of what constitutes one of the stylised facts but those that we look at – and probably the easiest to observe – are fat-tailed probability distributions and volatility clustering.

As stated earlier, the probability distribution function of the returns for financial data is often assumed to be Gaussian or Normal. In fact, this is one of the main assumptions of standard finance theory [60]. Real financial data, however, has been shown to produce fat-tailed return distributions [67]. The fat tail means that there is an increased probability of more extreme returns than would be predicted by a Normal distribution. This is one of the arguments for why Brownian motion is not a particularly good model of asset prices. For real financial data distributions the tails decay as a power-law. The value of the exponent for this has been shown to be -4 on average [68].

Clustered volatility is easy to spot in price series data. It is where volatile periods are grouped together, i.e., the regions of volatility are not randomly spread along the price series as would be predicted by standard finance theory but appear bunched up. Standard finance theory does not predict the presence of clustered volatility and

[1] Following from earlier notation, the stylised minority game is a member of the class of *exotic* minority games.

states that volatility is constant. Clustered volatility can be related to the number of trades taking place, or the number of traders in the market at any particular time [67]. This activity is measured by the volume, which is the number of contracts traded in a given period.

We use these two stylised facts as a test of the modified vanilla minority game.

16.2 Adaptation of the Minority Game to Produce Stylised Facts

One adaptation of the vanilla minority game to produce stylised facts is the Grand Canonical Minority Game (GCMG) [70]. This is a minority game in which the number of agents playing at each iteration is not fixed.[2] An agent will only participate in the game if he owns a strategy whose score is above some threshold. Another feature of the game is that the strategies' virtual scores are only recorded over a set number of timesteps. In our study we have used an alternative approach based on the GCMG [68, 58].

The first decision before modifying the vanilla minority game was the choice of virtual score reward. Based on the analysis of the effects upon the price series and return distribution of changing the reward, the reward given in equation 15.2 was chosen. As was shown, this reward resulted in significantly less emphasis of side peaks in the return histogram in the symmetric phase and the removal in the trending in the price series. It also showed some evidence of volatility clustering in the price series which was not evident when using the step reward.

To introduce the non-fixed number of participants to the game the agents are each given a bank account. The level of this account is incremented at each iteration by a constant amount, ϵ. An agent is only allowed to play the game if the virtual scores of at least one of their strategies is above the value of ϵ. This introduces the concept of agent confidence. An agent will only use a strategy if its past performance has been better than the risk free bank account in the same way as a trader may only enter the market if his trading strategies have been successful, giving him the confidence to trade. These agents or traders are called speculators. The total number of speculators is N_S and the number of active speculators is N_{Sact}.

Due to the symmetry in the scoring algorithm (see equation 15.2), strategies can be penalised by the same magnitude as they are rewarded, but because of the

[2]This idea was first proposed by M.Hart, P.Jefferies and N.Johnson. The nomenclature comes from statistical physics where grand canonical refers to a system with a non-fixed number of particles.

asymmetry in the game only a minority of strategies are ever rewarded, whereas the majority will be penalised. As the agents' bank accounts are being incremented by a constant positive amount at each iteration, while their virtual strategy scores are, on average, being reduced, over time the speculators will drop out of the game until none are playing, $N_{Sact} = 0$. This begs the question, if the minority game is a negative sum game then what is in it for the speculators? In order to correct this problem a second type of trader is introduced - the producer (N_P is the number of producers).

Unlike the speculator who comes in and out of the market trying to make money from his trading activity, the producer models a trader who uses the market as a mechanism for exchanging goods. The producer agents will trade at each iteration and hold only one strategy. Therefore their behaviour is deterministic with respect to μ. The result of the producers on the game is to introduce information, H, into the game for the speculators to exploit. Even though the overall minority game is still a negative sum game, the speculators, with their ability to adapt and bounded rationality, can exploit the deterministic behaviour of the producers and on average gain at each iteration. Hence, depending on the value of ϵ, the speculators can remain in the game, periodically moving in and out, depending on the performance of their own particular strategies.

The vanilla minority game can be modified with these two realistic properties to produce stylised facts. Firstly, it is reasonable to consider that real world markets are made up of traders using the market to exchange goods (e.g. asset managers), their actions adding information which can be exploited by a second group of traders that utilise this information to gain from the market movements whilst adding liquidity (e.g., proprietary traders). Also, it is realistic to think that these speculators will not always want to trade in the market, but will only enter if they feel the market conditions are such that they can make money. It is these speculators that add the stylised facts.

There has been further work on deriving parameter values (i.e., values of N_S, N_P, and ϵ) required to reproduce the stylised facts [69]. Also of note, this work states that even for the same parameters, some realizations of this modified minority game will produce stylised facts whereas in others they are absent. The only difference lies in the distribution of strategies among the agents at the beginning of the game. Further to this, if a second group of speculators are introduced to the game after some time then they can sometimes provoke stylised facts where there were originally none. This has been likened to traders following a stock for some period before entering the market to trade in it. In our study we have kept to just one set of speculators so as

to make the game easier to analyse, but the opportunity for different incarnations of the same minority game to arise has been considered.

16.3 Results for the Stylised Minority Game

Figure 16.1 shows a price series generated by the stylised minority game over 60,000 iterations. The most noticeable effect is the very large change in volatility at around

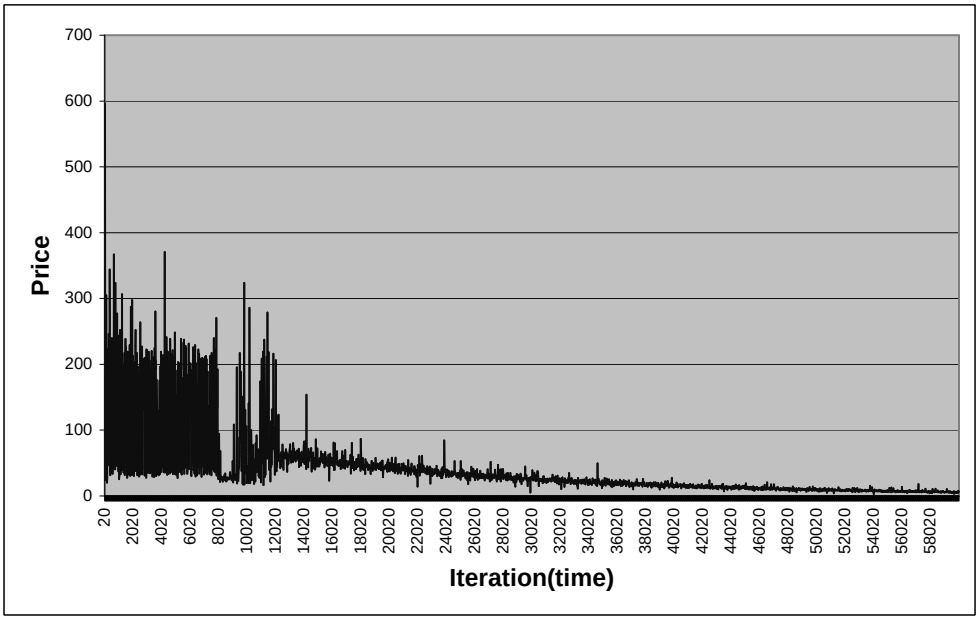


Figure 16.1: Change in price with time over 60,000 iterations for stylised minority game with $m = 3$, $s = 2$, $N_S = 501$, $N_P = 1000$, ϵ=0.01.

the 12,000 iteration mark. This is not present in the vanilla minority game and is due to the introduction of the trader confidence level. Initially the value of the internal bank account of each of the agents is such that it is easy for their strategies' virtual scores to beat it. As can be seen in figure 16.2, at first the number of active speculators in the game is large, comparable to the total number of speculators. After around 8000 iterations this drops fairly rapidly until at around iteration 12,000 it drops suddenly to an average value of around 50. From iteration 12,000 onwards there is a steady state where the increase in the value of the internal bank accounts, ϵ, matches the average increase in the virtual scores of the speculator's strategies. The higher the number of active speculators, the higher will be the volatility, as the price changes are dependent on the number of agents choosing each outcome and so more active speculators means a larger possible change in price can occur.

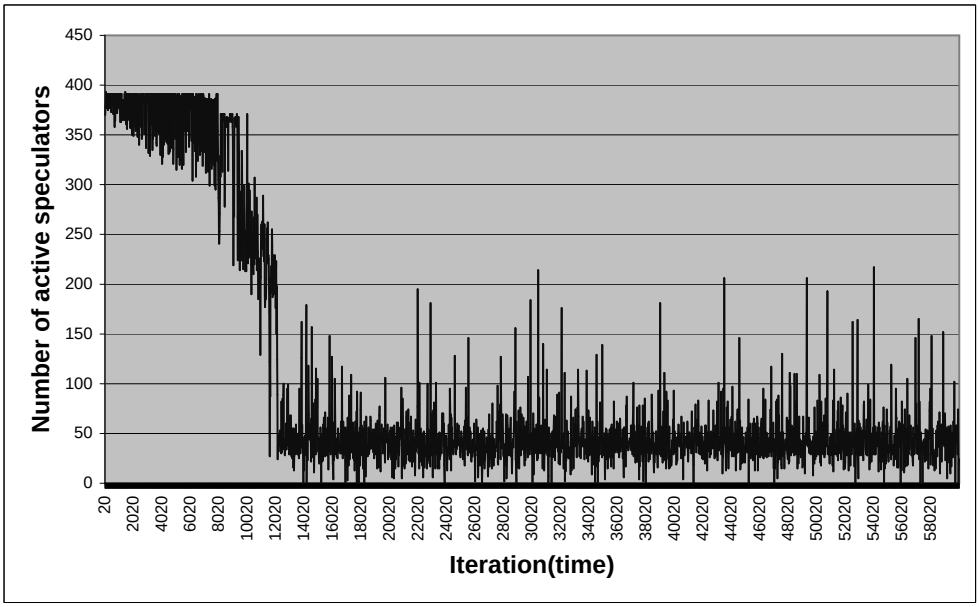


Figure 16.2: Change in the number of active speculators, N_{Sact}, with time over 60,000 iterations for stylised minority game with $m = 3$, $s = 2$, $N_S = 501$, $N_P = 1000$, ϵ=0.01.

Clustered volatility in the price series can be clearly seen in figure 16.3, which looks over 1000 iterations of the game. This may, in part, be the result of the

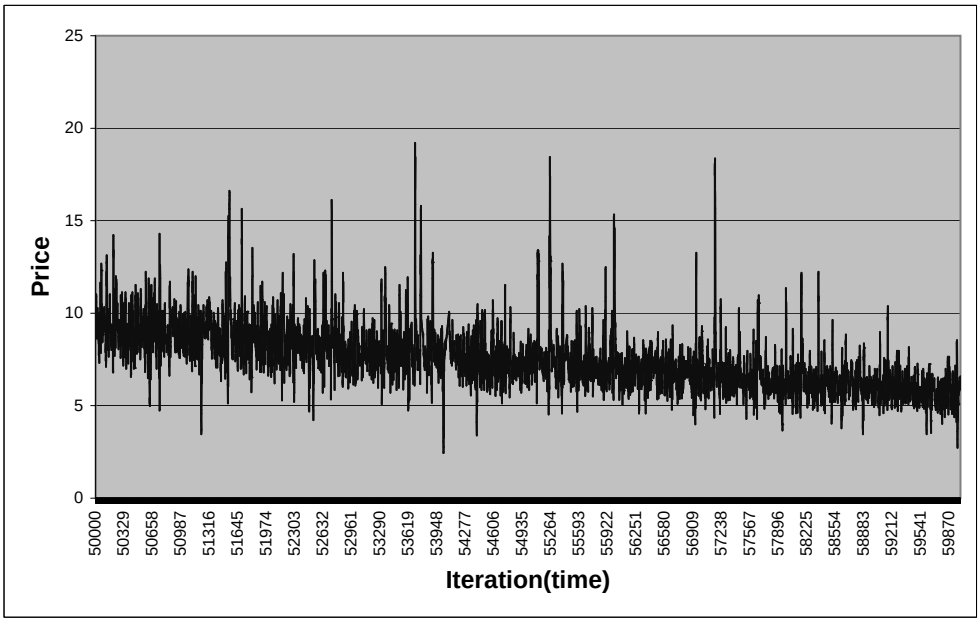


Figure 16.3: Detail from figure 16.1. Change in price with time over 10,000 iterations for stylised minority game with $m = 3$, $s = 2$, $N_S = 501$, $N_P = 1000$, ϵ=0.01.

clustering in the volatility of the number of active speculators, again see figure 16.2. As stated above, more active speculators will lead to larger price volatility. The volatility clustering in figure 16.3 is comparable to that in figure 15.6, which could lead one to think that at a deeper lever within the game the clustering is a result of the reward used to update the virtual scores. The clustering for the stylised minority game does appear more pronounced, but one must also keep in mind the warning from Challet referred to earlier, that different realisations of the game can lead to different degrees of stylised facts [69].

Figure 16.4 looks at just 100 iterations of a price series generated by the stylised minority game. Unlike the vanilla minority game, there is no obvious pattern to the

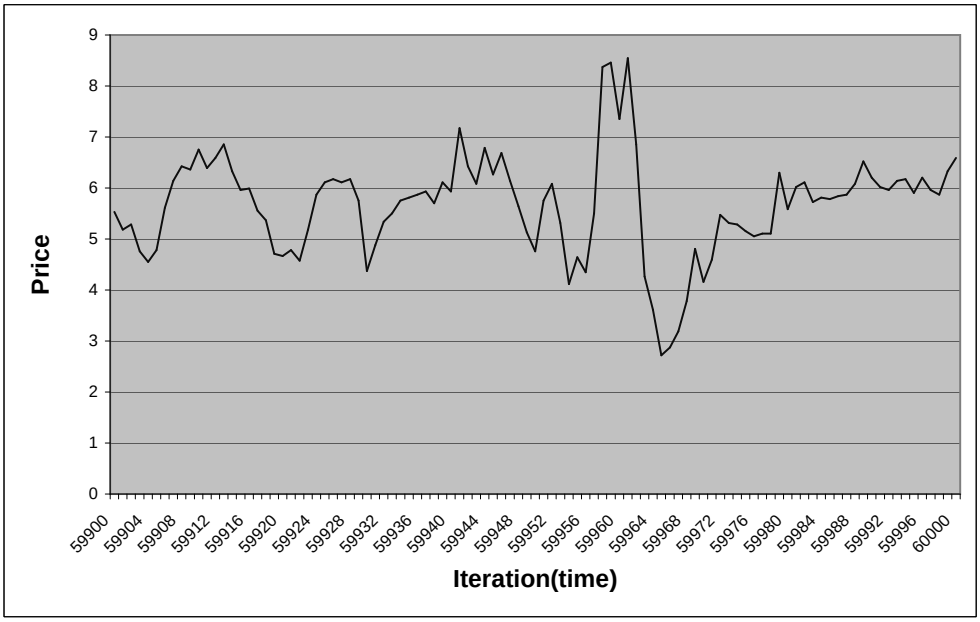


Figure 16.4: Detail from figure 16.1. Change in price with time over 100 iterations for stylised minority game with $m = 3$, $s = 2$, $N_S = 501$, $N_P = 1000$, ϵ=0.01.

price changes, and it is therefore more comparable to real world asset price dynamics.

The distribution of returns is shown in figure 16.5. The price series used to generate the returns was taken from iteration 12,000 onwards so as not to include the "settling" period of the game mentioned earlier. In contrast to the return distributions for the vanilla minority games this is clearly far closer to the Normal distribution but with fatter tails. In fact it has been shown that the return distribution for the stylised minority game exhibits an exponent of very close to the -4 value of real markets [68].

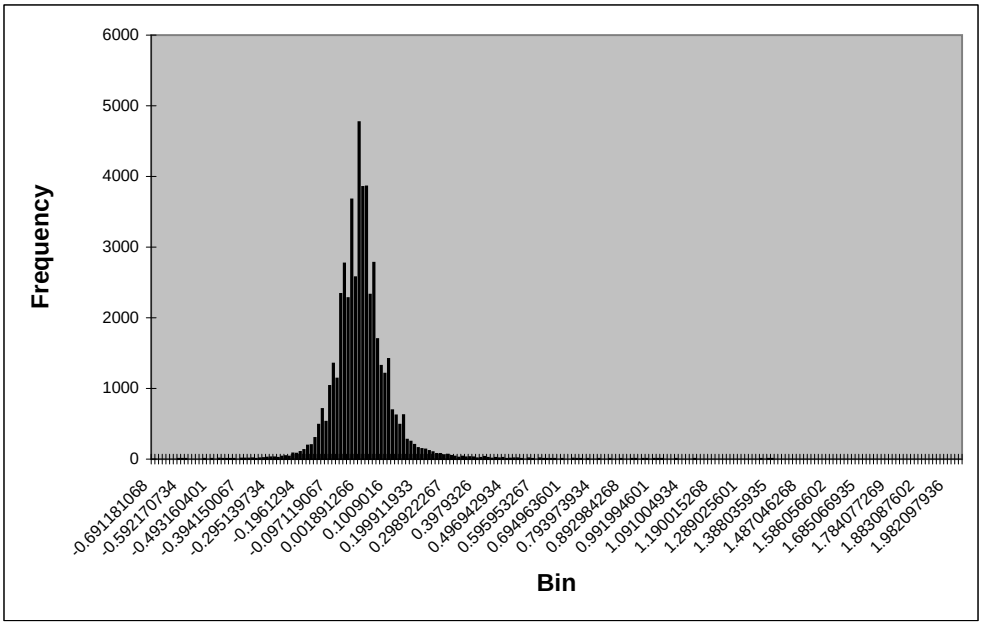


Figure 16.5: Histogram of returns for stylised minority game with $m = 3$, $s = 2$, $N_S = 501$, $N_P = 1000$, ϵ=0.01.

16.4 Limitations of the Stylised Minority Game as a Market Model

The stylised minority game is a significant improvement over the vanilla minority game as a market model. This is most evident in the return distribution, but also shown by the clustered volatility and non repetition of the price series over a small time scale. The mechanics behind the game are also more representative of the real world, with the addition of producers and trader confidence.

One limitation of the stylised minority game is the potential absence of stylised facts in some realisations of the game - something which is not studied here.

Also, it is possible to conceive of further modifications to the game to create a more realistic representation of real world market mechanisms. For example, even though the agents are heterogenous in terms of their individual strategies, they still share the same value of ϵ for their bank accounts. This is perhaps quite unrealistic as traders in the real world will have very different and varying confidence mechanisms.

However, without adding too much complexity to the vanilla minority game, the stylised minority game shows itself to be a reasonable representation of market dynamics whist still retaining the all important simplicity of any good model. It is therefore a good basis from which to study feedback effects of an option hedger.

Chapter 17

Option-Hedger Minority Game

In this chapter we first discuss feedback effects in markets using Black-Scholes analysis. We then add an option hedging agent to the stylised minority game and look at the feedback effects from hedging a variety of options over a range of parameters..

17.1 Feedback Analysis Using Black-Scholes

As well as buying and selling the option contracts themselves, a European call or put option can be replicated by trading in the underlying by an amount given by the delta of the option. This strategy can be used by portfolio managers (who often cannot use derivative products themselves) to insure against large movements in an asset. Some work has been done in the feedback effects of trading in an asset on the price of that asset. This may be through either delta hedging or option replication as they are effectively the same thing – delta hedging a short put position is the same a buying delta of a stock to replicate a put option. It is instructive to take a very brief look at some results of this work as an aid to the analysis and understanding of the minority game option hedge. These results are the creation of a modified price process to take account of feedback effects [71].

Assume that the excess demand for an asset is given by $\chi(S, t, x)$. It is a function of price S, time t, and a random influence x. This is analogous to the value $A(t)$ in the minority game. The equilibrium price $S_{Eq}(t)$ is the price for which supply is equal to demand, i.e. excess demand is zero:

$$\chi(S_{Eq}(t), t, x) = 0. \tag{17.1}$$

It is assumed that in any market there is a strong tendency to return to the equilibrium price if it has been disturbed, i.e., the market equilibrium is stable.

Now, add on to the excess demand function above a completely deterministic demand due to the trading strategy $\Delta(S,t)$ (for example, the portfolio replication mentioned above). The equilibrium condition becomes:

$$\chi(S_{Eq}(t),t,x)+\Delta(S,t)=0 \tag{17.2}$$

and hence the equation for the changes in χ and Δ is:

$$d\chi+d\Delta=0. \tag{17.3}$$

Assume that the excess demand function can be expressed as

$$\chi(S,t,x)=\frac{1}{\psi}(x-S) \tag{17.4}$$

where ψ is a positive real number and x follows the stochastic process

$$dx=\mu_x dt+\sigma_x dX. \tag{17.5}$$

Therefore, x can be thought of as the "intrinsic" value of the asset.

Then ψ is a measure of how strongly the excess demand function reacts to price changes. A price change of dS results in a change in the excess demand of $-dS/\psi$, i.e., ψ is indicative of the liquidity of the market. In a liquid market there is a strong reaction in excess demand to a change in price and hence ψ is small. For illiquid markets ψ is large as a change in price does not result in a significant change in demand for the asset.

The equilibrium condition in equation 17.2 can now be written as:

$$(x-S)+\psi\Delta(S,t)=0. \tag{17.6}$$

Applying Itô's lemma (see [60]) to equation 17.6, the stochastic process followed by S is:

$$dS=\mu_S dt+\sigma_S dX \tag{17.7}$$

where

$$\mu_S=\frac{\sigma_S}{\sigma_x}\left(\mu_x+\psi\left(\frac{\partial\Delta}{\partial t}+\frac{1}{2}\sigma_S^2\frac{\partial^2\Delta}{\partial S^2}\right)\right) \tag{17.8}$$

and

$$\sigma_S=\frac{\sigma_x}{1-\psi\frac{\partial\Delta}{\partial S}}. \tag{17.9}$$

If one chooses $\mu_x=\mu S$ and $\sigma_x=\sigma S$ then equation 17.7 becomes geometric Brownian motion when $\psi=0$ [71].

For the study of the option hedger minority game the most interesting result in the above analysis is shown in equations 17.8 and 17.9. Both the drift μ_S and the volatility σ_S have the denominator:

$$1 - \psi \frac{\partial \Delta}{\partial S}. \tag{17.10}$$

Therefore, the value of the drift and volatility, taking into account feedback from a trading strategy Δ, is dependent on the rate of change of delta with the price of the asset – its gamma. A positive gamma,[1] results in an increase in the drift and volatility of the asset. Therefore, for an option with positive gamma, delta hedging a short option position or replicating a long option position destabilises the market of the underlying. Increasing the value of gamma increases the destabilising effect of the feedback.

17.2 Setup of the Option Hedger Minority Game

The option hedger minority game is the stylised minority game with an additional agent that behaves in such as way as to simulate hedging a short option position. Here, we look in detail at the setup of an option hedging agent hedging a short European call option.

At the beginning of the game the option hedging agent is given the iteration at which it is to start hedging the option. As shown earlier when looking at the stylised minority game, this iteration needs to be large enough to give the game time to stabilize. The option maturity, T, is the number of iterations after the start iteration at which the contract matures. In fact, in all the hedging calculations time is measured in terms of iterations. The other parameters the option hedger is given are the:

Strike, E - This is the strike of the option and is usually input as a level above or below the current price. This is because after many iterations of the game the price may randomly be a long way from the start price.

Number of options to hedge, N_{opt} - The option hedging agent can hedge a position that is short more than one option. This changes the impact the option hedger has upon the market.

[1] A positive gamma means that the payoff profile of the option is concave. This is the case for replicating long put and call positions and hence hedging short put and call positions.

Risk free rate, r - This is risk free rate used in the calculation of delta. For self consistency, , this can be set to the same value as the risk free rate on the speculators bank accounts, ϵ, or given a different value depending on the parameters of the game.

Volatility of the underlying asset, σ - This can be calculated individually for each run of the game by generating a return series from the minority game price series. Or, it can be set to be the same for many runs of the game on a value based on real world markets.

Number of iterations between re-hedging, f_{rh} - The option hedging agent can rehedge its short position every f_{fh}th iteration.

On the first iteration in which the European call option contract becomes active the option hedger will need to buy a quantity delta of the asset so as to remain delta neutral. Using equation 14.7 the option hedger calculates the value of delta based on the input values of E, r, σ, T, and the values S and t from the game. S, the price of the underlying asset, is taken from the previous iteration of the game. As mentioned earlier, the time within the life of the option, t, is measured in number of iterations. From the value of delta the hedging agent calculates how much of the underlying asset it must buy. For the option hedging agent to buy one unit of the asset, then the number of agents predicting a 1, or wanting to buy, is increased by one. Therefore, at the beginning of the contract, if the delta of the option is calculated as 0.72 and the option hedging agent is hedging 10 options, then the number of agents choosing 1 for that iteration of the game is increased by 7. Note that if there is only one option to hedge then due to rounding the effect of the option hedger will be to either buy or sell one unit depending whether delta lies less than, or above or equal to 0.5.

The option hedging agent keeps track of how much of the asset it currently holds as a hedge. If, on the second iteration of the contract life, the price of the underlying changes such the delta is reduced, then the option hedging agent will need to sell some of its hedge. This will translate as an increase in the number of agents predicting 0, for that iteration.

This activity of buying and selling is repeated throughout the life of the option depending on the changing value of delta, but not necessarily at every iteration.

17.3 Results for Option Hedger Minority Game

17.3.1 Hedging a Call Option

Figure 17.1 shows the change in the price of the underlying asset and the delta of the European call option over the life of the option contract. The parameters chosen are the result of much trial and error. We wanted to find a balance between setting the parameters equal to those values found in real world markets and values that produce clear results. Figure 17.2 shows the beginning of the life of the option in figure 17.1.

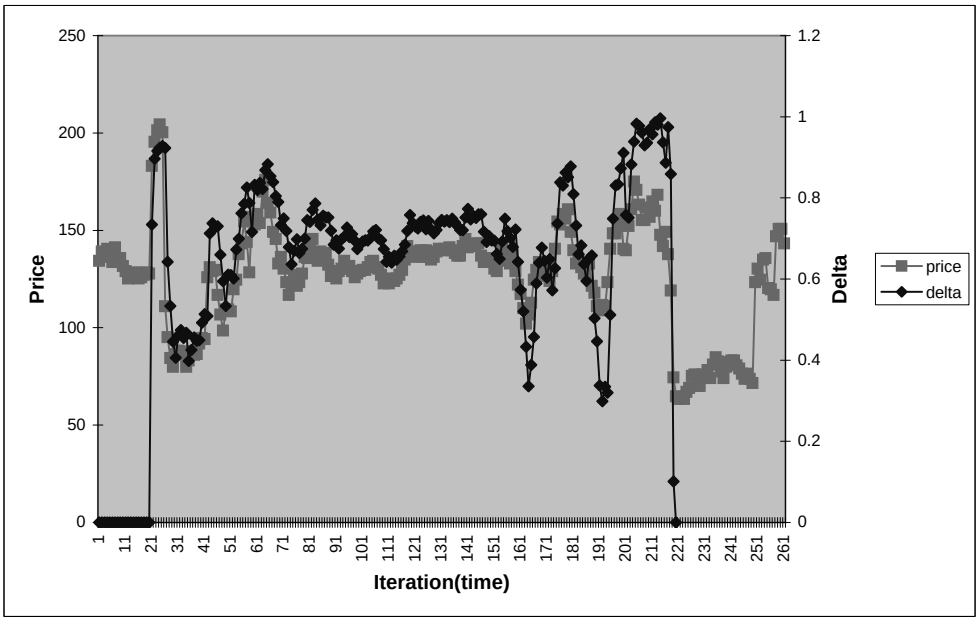


Figure 17.1: Change in price and change in delta with time, hedging a vanilla call option. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 128.512$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

By studying many runs of the game (see figure G.3) a clear pattern to the initial behaviour of delta and the asset price emerges.

Initially, the option hedging agent needs to buy a large number of the underlying asset to hedge the short call position. The size of the buys and sells of the option hedger can be seen in figure 17.3. The large buy drives up the price of the underlying asset - at this point the change in delta is leading the change in price. Then, due to the rise in price at the previous iteration, the delta of the option increases - the price is now leading delta. This has the effect of pulling up the price of the underlying asset a little more. As can be seen in figure 17.3, there is significantly less asset bought by the option hedger at the second iteration of the contact life, hence a much smaller

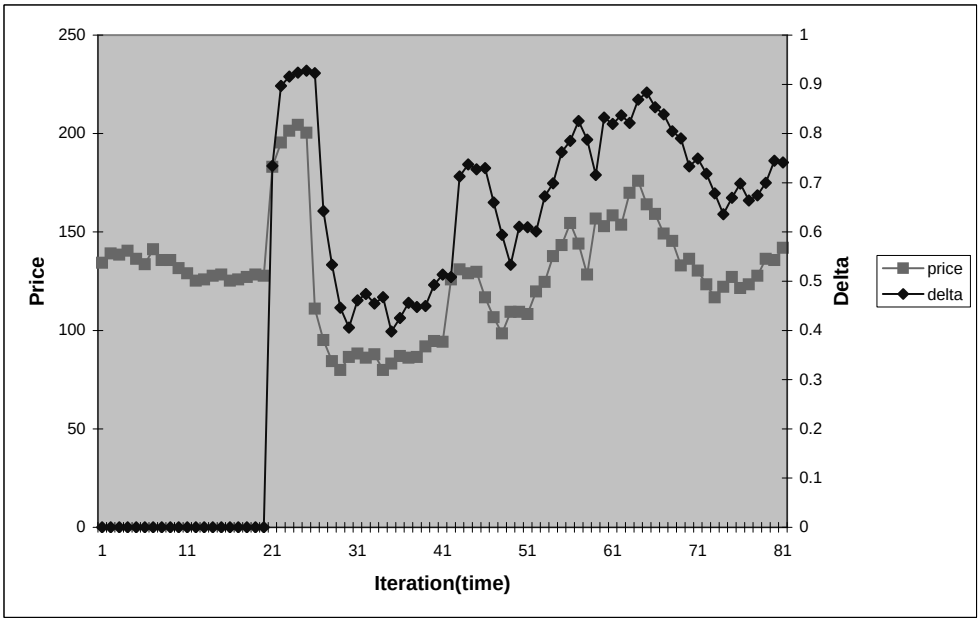


Figure 17.2: Detail from figure 17.1 showing start of option contract. Contract starts on iteration 20.

change in price. Looking at the scale on the right hand side of figure 17.2, the value of the option delta is now very close to one. The shape of the call option delta can be seen in figure 17.4 and the shape of the call and put option gamma is shown in figure 17.5. Over the following few iterations of the game the value of delta and asset price remain high until a sudden drop in the asset price results in a drop in the value of delta. We come to the reason for this sudden drop in asset price later. This drop in price and subsequent drop in the value of delta, results in the option hedging agent selling some of the underlying asset (again see figure 17.3). This means there is a further drop in price and again a further drop in delta. This pattern would repeat indefinitely if not for the shape of the option delta.

The rate of change of delta (gamma) is at a maximum when the delta of the option is equal to 0.5. Hence, at this value of delta one can expect to observe the largest contribution from the option hedger to the asset price change. As the delta of the option moves toward one or zero, the contribution to the asset price change reduces until its effect upon the asset price is insignificant. If the delta is near zero, it is only when the the asset price starts to increase that the option hedger then needs to adjust its delta hedge and thence can the adjustment to this hedge move the price up further. As the delta moves towards 0.5 its contribution is at a maximum and the price will continue to rise until the value of delta is close to one. This pattern continues

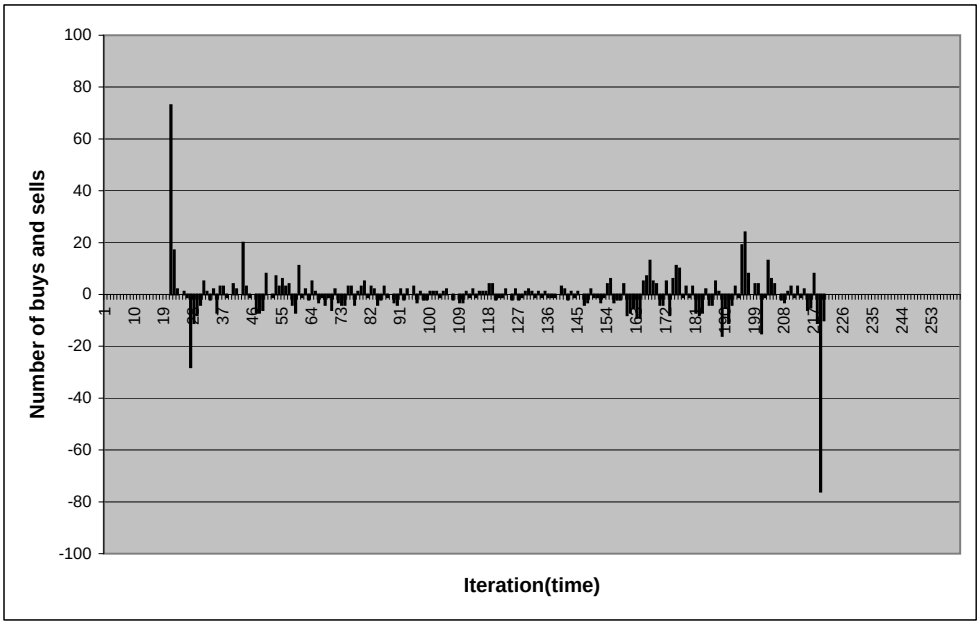


Figure 17.3: Number of buys and sells of option hedging agent over life of contract. Positive is buy, negative is sell. From same run as in figure 17.1. Contract starts on iteration 20.

throughout the option life, giving rise to a oscillatory behaviour to the underlying asset price. This can be clearly seen by referring back to figure 17.1. Towards the end of the option life there is a movement of the asset price that is tracked by a oscillation in the value of delta between near zero and one. Even though the delta hedging is responding to changes in the value of the underlying, its very presence dictates the overall movement of this asset price.

As mentioned earlier, there can occur very large changes in the asset price which are not explained by the above analysis. If one looks back at the first few iterations of the game (figure 17.2), after the initial sharp rise in the asset price, caused by the option hedger first hedging its position, there is then a sharp drop again in the price. This is due to the mechanics of the minority game itself.

The table in figure 17.6 shows the change in price over the first iterations of the option life accompanied by the minority outcome for each iteration. By looking back over the previous m outcomes (here $m = 4$) one can see the history that preceded a particular change in price. The first two figures in bold in the left hand column are the prices before and after the option hedger first becomes active, driving up the price of the asset (note, the actual values are for the run shown in figure G.3). The size of the buy made by the option hedging agent is such that all the virtual scores

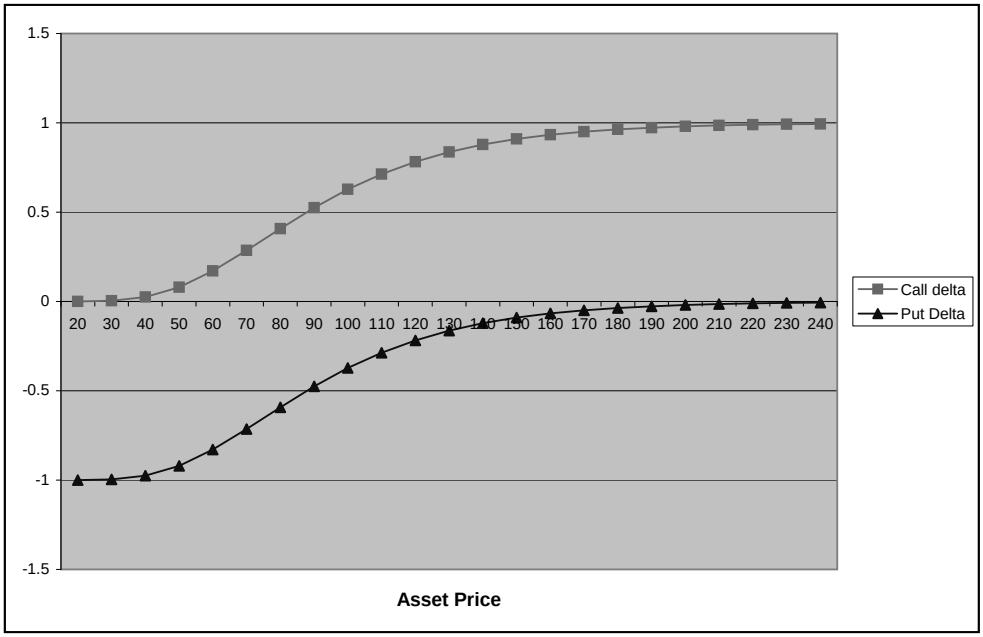


Figure 17.4: Plot of delta for European call and put options with strike of 100.

of strategies that correctly predicted the minority outcome are heavily rewarded, see equation 15.2. It does not matter that this may have not been the minority outcome had the option hedger not been active. Due to the large increase in virtual score, these strategies then become the best performing strategies for those agents that own them, and hence the ones that they will use. The random creation of strategies at the beginning of the game means that even though they all predicted the same outcome for the history that led to the first large jump in price, on average half will predict one outcome, half the other outcome, for the other histories that occur. Therefore, after the initial rise, for a few iterations at least, there is comparably little change in price as the action of the agents is balanced. Now, when the history that led to the large price rise occurs again, all the agents that predicted the first outcome of 0 to be in the minority, will do so again. However, this time 0 will not be the minority outcome and all those agents will be penalised. The price will suffer a large drop as so many agents will be predicting sell. This can be seen in the second set of bold figures in the left hand column of figure 17.6 and the corresponding history in the right hand column.

As the other agents in the game have no knowledge of the option hedging agent, by rewarding the strategies at the beginning of the game by such a large amount, whatever history preceded this is now "loaded". The option hedger making such a large buy will force the minority outcome to 0 or sell. The strategies that did not

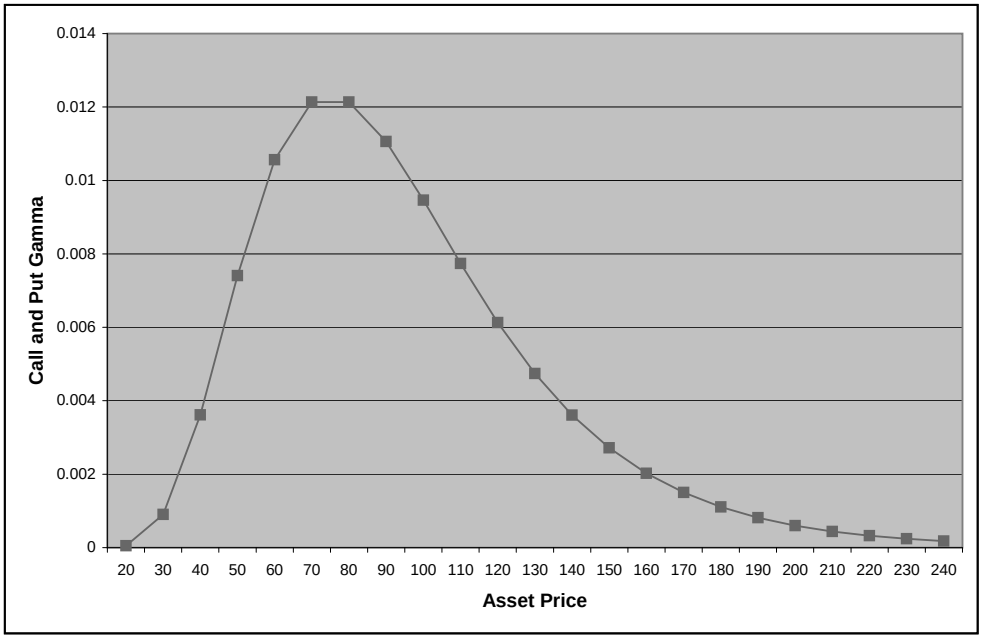


Figure 17.5: Plot of gamma for both European call and put with strike of 100.

predict the correct minority outcome on the first turn of the option hedging agent are penalised heavily and unlikely to be used in the near future. The agents continue to play until the "loaded" history repeats itself. Then, those *majority* of agents that own, and hence are using, strategies that were rewarded initially, all predict the same outcome of 0 or sell. The large number of agents selling will drive the price of the asset down. The virtual scores of those strategies are penalised while those that were at first penalised are now rewarded. Therefore, even though this history "loading" effect from the initial price rise can and sometimes does continue occurring further into the game, its magnitude will become damped as the strategy virtual scores come together, i.e., if that particular history were to occur again, even though the same group of strategies would all predict the same outcome, they may no longer all be the agent's best performing strategies, and hence the ones they would use.

As an aside, it is interesting to note that after the initial jump in price at the start of the contract, the length of time before the price drop can be dependent on this history that occurred before the first jump. For example, if the history was 0101, as the next outcome will be 0 due to the action of the option hedger, then it only needs for the next outcome to be a 1 and already history has repeated itself and there will be a drop in price. However, if the history was 1111, then as the next outcome is zero, it could be a long time before 1111 reappears. The price may have drifted

Price	History
144.051	0
145.499	0
123.368	**1**
129.693	**0**
147.698	**0**
163.232	**0**
265.117	0
317.402	0
273.191	1
261.17	1
265.117	0
267.781	0
238.691	1
237.501	1
235.137	1
237.501	0
215.977	1
183.125	1
152.959	1
151.437	**1**
160.801	**0**
175.067	**0**
184.966	**0**
103.562	1
96.5605	1
93.7067	1
102.532	0
107.788	0

Figure 17.6: Table of price with game outcome. From same run as in figure G.3.

downwards due to the shape of the option delta by then. This leads on to the effect of changing the memory of the agents, which is looked at later in the report.

Towards the end of the option life ($t \to T$), the delta of the call option will become steeper around the strike. This manifests itself as more frequent shifts in the asset price as delta responds more strongly to any change in price. This can be seen in figure 17.1.

This analysis relates well to the previous analysis of feedback using Black-Scholes. Firstly, it is clear that the presence of the option hedger increases the volatility of the underlying asset (compare figures G.1 and G.2). This is in agreement with equation 17.9, which predicted that the presence of a trading strategy Δ to replicate an option with positive gamma (the same as hedging a short position is a option with positive gamma), results in a increase in market volatility.

Further to this, as the gamma is increased, i.e., as we move towards the end of the option life and the delta becomes steeper, then the instability of the underlying asset seems to increase - the frequency of shifting between high and low delta increases. Therefore, as stated earlier, increasing the value of gamma appears to increase the destabilising effect on the market.

At the very end of the option life the delta will move to either one or zero, de-

pending which side of the strike price the asset price lies. This large move in delta will provoke a large price change, effectively reinforcing the final hedge position by moving the price further from the strike. Again, this large price change can cause a "loading" effect which results in a large opposite price change back towards strike price *after* the option contract has matured.

17.3.2 Change in Agent Memory

Up to now, the memory of the agents in the stylised minority game, used to produce the market in which the option hedger acts, has been 4. Figure 17.7 shows the beginning of the contract life with $m = 2$. The most striking difference to the previous results is an increase in the volatility of the option delta and an increase in the frequency of movements between high and low delta. The main reason for this appears

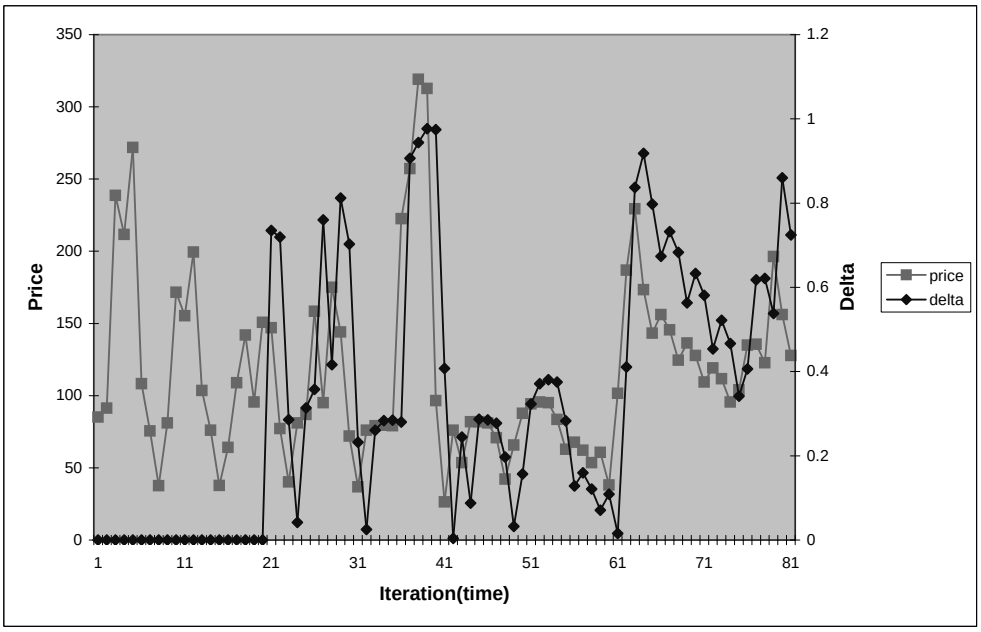


Figure 17.7: Change in price and change in delta with time, hedging a vanilla call option. Contract starts on iteration 20. $T = 200$, $m = 2$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $k = 151.432$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

to be the increase in price volatility in general, with or without the option hedger present. By comparing figures G.4 and G.5, it is clear that the reduction in the agent memory produces a more volatile market environment. This is in line with the results from the study of the vanilla minority game and figure 15.2. However, the presence of the option hedger does seem to increase this volatility slightly. This is due to the

feedback effect of the option hedger agent on the price, adding to every large price jump at the next iteration with a further move in the price in the same direction.

Also, the large changes in price that occur will cause more instances of history "loading" throughout the game. As the history string in now much shorter there should be, on average, a much shorter time before history repeats itself. This would explain the increase the frequency of the movements between high and low delta.

Such a volatile market generated by $m = 2$ is perhaps fairly unrepresentative of real world markets. Hence the initial choice of $m = 4$ which, to the eye at least, provides us with a more realistic market model.

17.3.3 Change in Number of Options to Hedge

The number of options (N_{opt}) hedged up to now has been 100 for 501 speculators and 1000 producers. Figure 17.8 shows the effect on the market of an option hedging agent hedging a position of only 10 options. Even with the initial jump in delta from 0 to around 0.7, there is little, if no impact of the option hedging agent on the price. All the adjustments to delta that track the price moves are so small that they produce no further moves in price. This can be seen by the size of the buy and sells of the option hedger in figure 17.9. Compared to the number of buys and sells made by the option

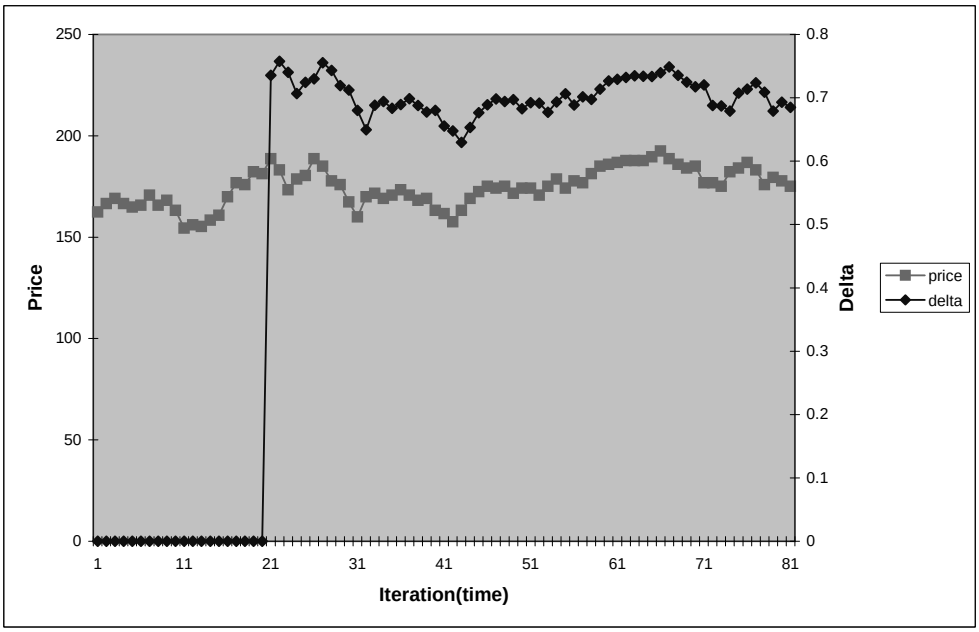


Figure 17.8: Change in price and change in delta with time, hedging a vanilla call option. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 182.053$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 10$.

hedger for $N_{opt} = 100$ (see figure 17.3), the actual buys and sells are very infrequent, and apart from the initial buy, are not greater than 1. The buys and sells are clearly not often enough nor large enough to sway the minority outcome generated by the speculators and producers. This demonstrates that the option position does have to be of a certain size before it impacts the market, which it what one would expect.

Figure G.6 shows the effect of an option hedger with $N_{opt} = 25$. Now, the initial buy from the option hedger is enough to move the market upwards and then produce another up move in the price as delta adjusts itself to the first price rise. Even though the buys and sells of the option hedger after this are not much larger than those for $N_{opt} = 10$, they are much more frequent (see figure G.7). This, along with the effect of history "loading" would explain the increase in volatility of the market created by the option hedger, which is not present for $N_{opt} = 10$. If one looks at figure G.7, one can see an increase in the size of the buys and sells as one moves towards maturity of the option contract. As stated earlier, this is the effect of the increase in the option gamma and hence the steepening of the option delta at the end of the contract.

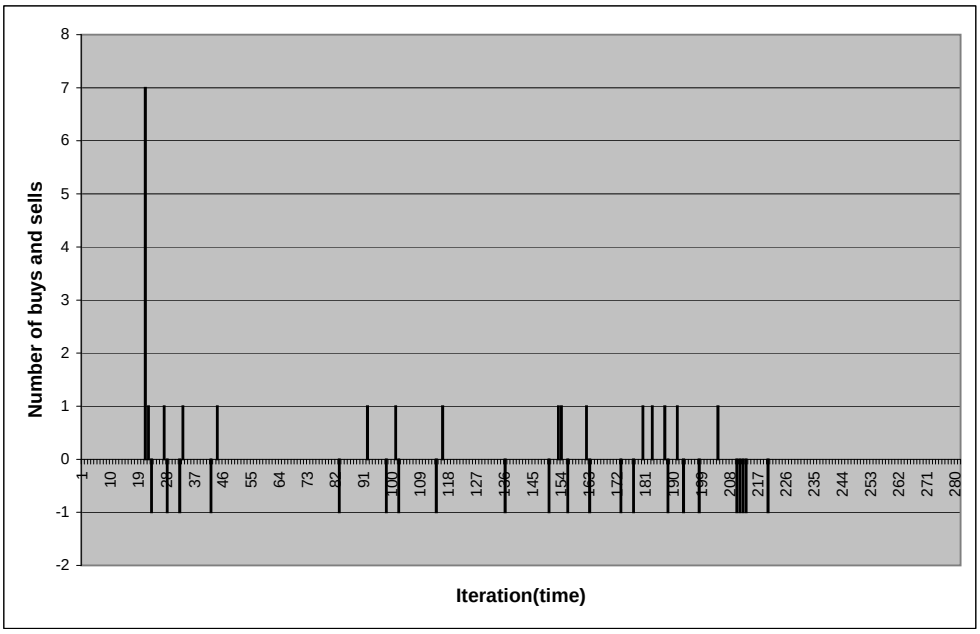


Figure 17.9: Number of buys and sells of option hedging agent over life of contract. Positive is buy, negative is sell. From same run as in figure 17.8. Contract starts on iteration 20.

17.3.4 Hedging a Put Option

Figure 17.10 shows the feedback effect on the market of an option hedger hedging a European put option. The most noticeable difference compared to the market under a call option hedger is the initial drop in the market price caused by the option hedger making a large *sell* to hedge its position. At first, the behaviour of the market under the put option hedger is opposite to that under the call option hedger, before settling down into the quasi periodic motion seen previously.

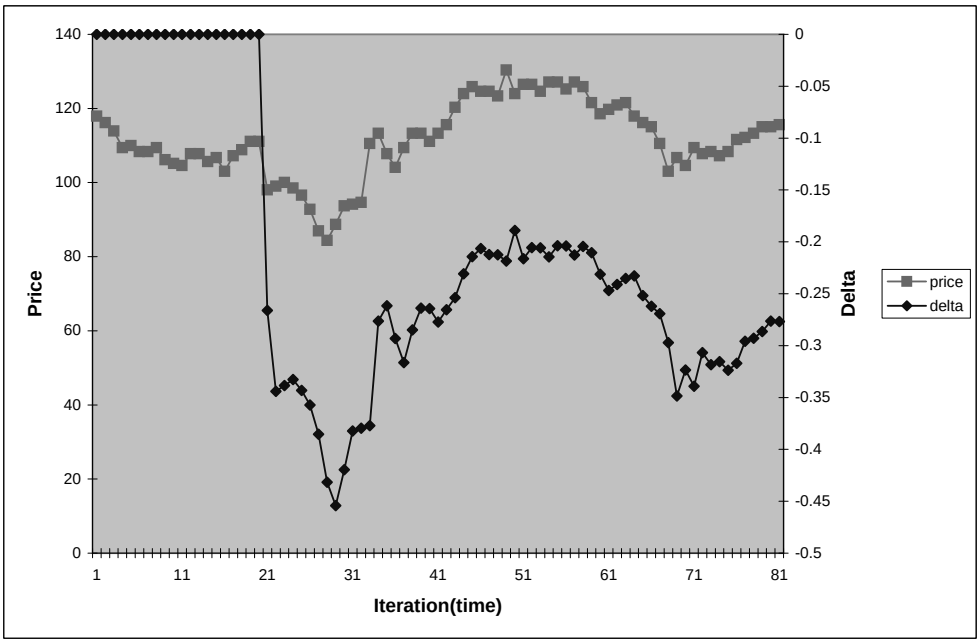


Figure 17.10: Change in price and change in delta with time, hedging a vanilla put option. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 167.279$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

Figure G.8 show the number of buys and sells of the option hedger. Again, there is a clear increase in the option hedger activity toward maturity.

17.3.5 Re-hedging Every nth Iteration

Up to now, the option hedger has been re-hedging at every iteration. In the real world, market liquidity constraints may make re-hedging this often prohibitively expensive. Figures 17.11 and G.10 show the response of the market to an option hedger re-hedging every $25th$ iteration.

Now the option hedger cannot follow a large price change with another push on the price in the same direction at the next iteration. It must wait 25 iterations before

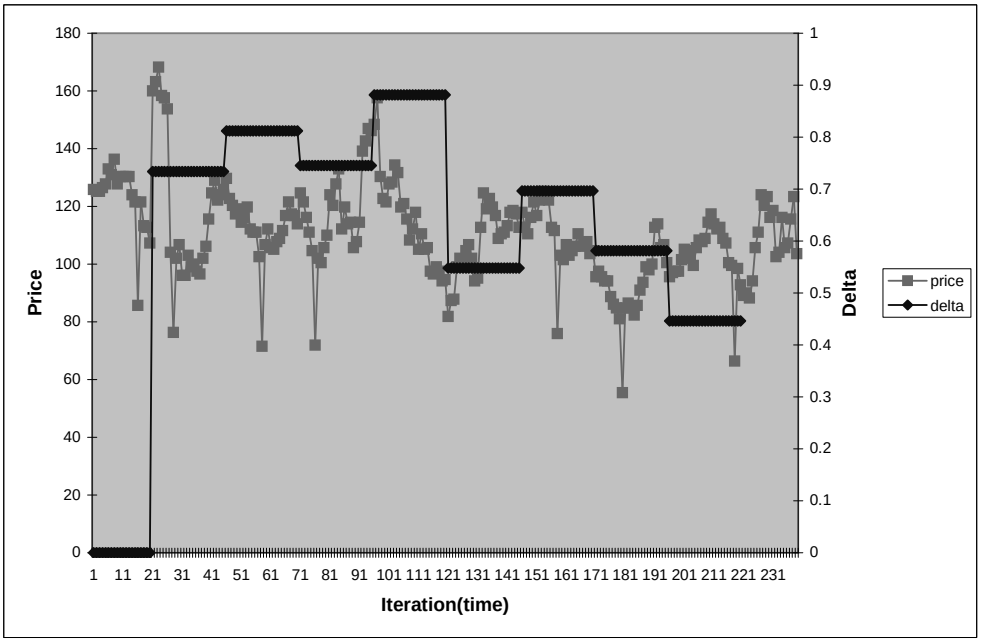


Figure 17.11: Change in price and change in delta with time, hedging a vanilla call option. Re-hedging every 25th iteration. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 108.001$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

responding, enough time for the price to have moved back in the opposite direction and for the large change to the hedge position to become unnecessary. This could act to reduce market volatility when compared to re-hedging every iteration. By the same token, however, the price can move further away from the value where the hedge was last set, meaning an even larger rehedge is eventually required - increasing market volatility. The combination of both of these actions seems to produce a market volatility comparable to when re-hedging every iteration. The main difference is the lag between the change in price and the response of delta. The longer that is left between re-hedges will reduce the frequency of large price changes as the delta hedger is acting less often.

Figure G.11 shows the results for a run of the game where the option hedger is re-hedging every $10th$ iteration. The option hedging agent is having a feedback effect on the market, but this time delayed by 10 iterations. Again, the same arguments as above should apply. The delay in the buys and sells of the option hedger, in response to the price changes, can be seen in figure G.12. The frequency of large changes in price is visibly greater than for re-hedging every $25th$ iteration.

17.3.6 Hedging a Binary Call and Put Option

A binary or digital option, is an option with a step function payoff at expiry. At maturity a binary call (put) contract pays one unit if the asset lies above (below) the exercise price. The binary call option value is given by:

$$e^{-r(T-t)}N(d_2) \tag{17.11}$$

where d_2 is given by equation 14.5. Plots of the delta for call and put binary options are shown in figure 17.13.

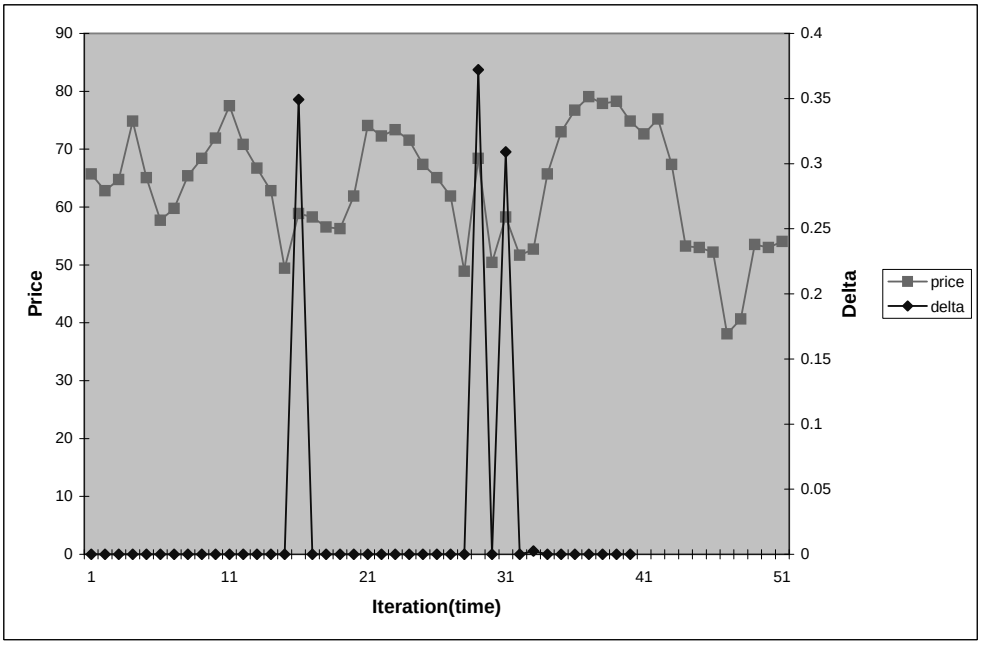


Figure 17.12: Change in price and change in delta with time, hedging a binary call option. Detail for end of option contract life. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 50.1576$, $r = 0.001$, $\sigma = 0.004$, $N_{opt} = 100$.

Figure 17.12 shows the feedback effect on the market of an option hedger hedging a short position in a binary call option towards the end of the contract life. As the market price passes through the option strike the option hedger acts driving up the price of the market back away from the strike. Now the market price is safely above the strike, the option hedger removes its hedge, by making a sell, and pulling down the market price again. If the price is pulled down below the strike again, then this pattern of events will repeat. By hedging the short option position, the option hedger is actually stopping the market price move below strike. The shape of the option delta is such that it is zero if the price is well above or below the strike. It is only around the option strike that the delta increases.

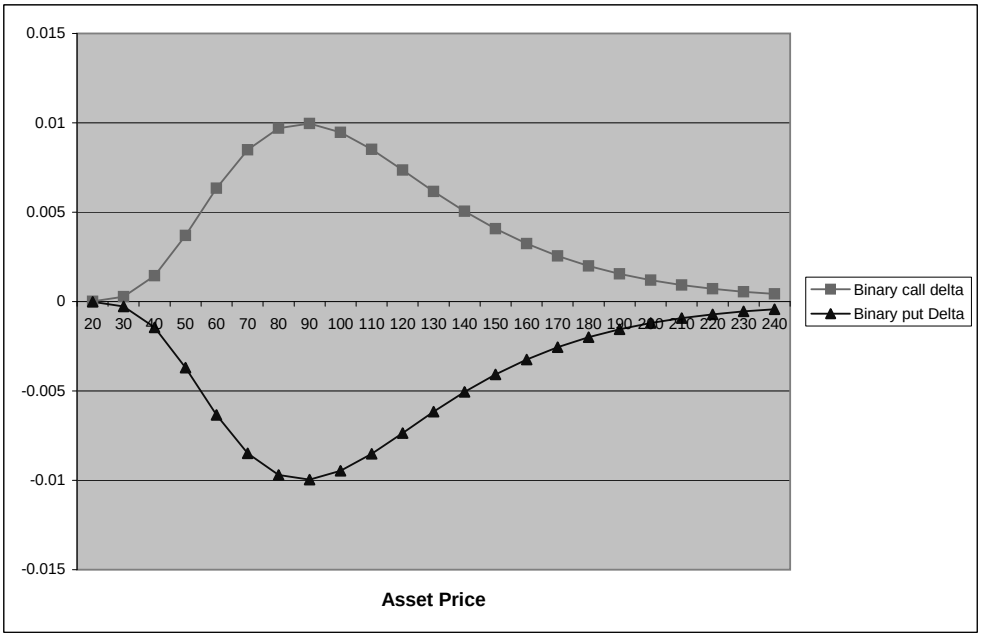


Figure 17.13: Plot of delta for Binary call and put options with strike of 100

This effect becomes more pronounced as the contract moves towards maturity (in the limit $(T-t) \rightarrow 0$ the delta becomes a *Dirac delta* function). Therefore, as one moves towards contract maturity the spikes in delta increase as its response to a price change around delta increases. Figure G.15 clearly demonstrates the increase in the activity of delta towards the end of the contract life. The narrowing of the delta function should also mean the underlying price should need to be closer to the exact value of the strike before the option hedger will act, but perhaps due to the discrete nature of the game this effect has been difficult to spot.

Chapter 18

Conclusions and Discussion

Two mechanisms dominate the study of the effects of dynamic hedging on the price of an underlying whose value is modelled by the minority game. The first is the feedback effect of the option hedging agent upon the price. This nature of the feedback is dependent on how the option delta varies with asset price – its gamma. For European options there is positive feedback, with the option hedger adding to any market price changes. For binary options the feedback is negative, the option hedger initially acting to move the price back away from strike. These feedback effects one may expect to find in real world markets, but only for very large option positions or the replication of very large portfolios, or for very illiquid markets.

The second effect is the large changes in price as a result of the minority game becoming "loaded" for a particular history, i.e., a large market change occurring after one set of events causes the agents to act the same when that set of events occurs again - when history repeats itself. This effect is harder to associate directly with real world markets. However, one may liken it to a situation where traders who trade based on historical data, act as a group when the history that preceded a crash or rally presents itself again, causing another large price movement. In the minority game this movement is in the opposite direction to the previous large change.

A minority game of low memory agents increases the volatility of the market. One may tentatively say that when traders in the real world shorten the horizon over which they view past events, this produces an environment of increased volatility. For example, shortly after the occurrence of a crash.

As the number of options to hedge is reduced, the impact of the re-hedging on the underlying also reduces. There is a point where the option hedger has no effect at all. This is probably in fact the case for most option positions in the markets, only very very large positions showing any feedback at all.

Re-hedging less often acts in such a way as to both increase and decrease the volatility of the market at different points, dependent upon the movement of the underlying. This is a result of the lag between the feedback effect from the delta and the changes in price. Liquidity constraints in real markets may demand infrequent re-hedging, and also, due to the reduced liquidity, increase feedback effects.

Overall, the stylised minority game proves to be a good platform on which to study feedback effects in markets. The simplicity of the market model allows one to distinguish between effects produced by the model's mechanism and the effects from market mechanisms. Most importantly however, the response of the market model is consistent with both calculation *and* intuition of real world markets.

It would be interesting to include in the option hedger model a measure of the transaction costs of trading. This could lead to the development of optimal trading strategies which would take into account the feedback effect produced by the hedger.

Also, including other types of option contract, maybe those with the possibility of early exercise, or whose value is path dependent, such as Asian options.

More study is required into the effects of not hedging at every iteration of the game as the current set of results is somewhat inconclusive. This would link in well with the study of transaction costs above.

Using a more sophisticated stylised minority game may produce interesting results. As mentioned earlier, one can use multiple sets of speculators introduced at various stages of the game. Perhaps, also using multiple option hedgers, hedging contracts with different maturities, volatilities, start dates etc. This would create a more representative model of real world markets than the current model of just one option contract at any one time.

Chapter 19

Overall Conclusions and Discussion

In the thesis we have looked at dependent defaults from two different perspectives; a reduced-form intensity based approach, and an agent-based approach. Both of these approaches use ideas from graph theory and complex network analysis to understand the structure of the dependency between the firms. We have also looked at agent-based systems from two different perspectives; the agent models of default, and a minority game based study of feedback effects from dynamic option hedging.

Overall, we feel that this study into Credit Networks and Agents Games has been a successful attempt to understand a little more about the default dependency between firms, and also the role of agent-based simulations in modeling the financial markets.

Appendix A

Poisson Processes

A.1 Definition of a Poisson Process

Consider a process of point events where $N(t, t+\Delta t)$ denotes the number of events in the interval $[t, t+\Delta t]$. Then, for some positive constant ρ, as $\Delta t \to 0$,

$$\begin{aligned} \text{prob}\{N(t,t+\Delta t)=0\} &= 1-\rho\Delta t+o(\Delta t), && \text{(A.1)} \\ \text{prob}\{N(t,t+\Delta t)=1\} &= \rho\Delta t+o(\Delta t), && \text{(A.2)} \\ \text{prob}\{N(t,t+\Delta t)>1\} &= o(\Delta t), && \text{(A.3)} \end{aligned}$$

and $N(t, t+\Delta t)$ is independent of occurences in $[0, t]$, see [29]. A process satisfying the conditions above is a Poisson process of rate ρ.

A.2 Basic Properties of a Poisson Process

Take a new time origin at t_0, which can be any fixed point including a point at which an event has just occured. Any property of the process referring to its behaviour after t_0 is independent of what happens at or before t_0. Therefore, if $t_0 + Z$ is the time of the first event after t_0, then the random variable Z is independent of whether an event occurs at t_0 and of any occurrences of events before t_0.

The probability distribution of the random variable Z is given as follows. Let

$$P(x) = \text{prob}(Z > x). \qquad \text{(A.4)}$$

Then, for $\Delta x > 0$,

$$\begin{aligned} P(x+\Delta x) &= \text{prob}(Z > x+\Delta x) && \text{(A.5)} \\ &= \text{prob}\{Z > x \text{ and no event occurs in} \\ &\quad [t_0+x, t_0+x+\Delta x]\} && \text{(A.6)} \\ &= \text{prob}(Z > x)\ \text{prob}\{\text{no event occurs in} \\ &\quad [t_0+x, t_0+x+\Delta x] | Z > x\}. && \text{(A.7)} \end{aligned}$$

A special feature of the Poisson process, a form of the Markov property, is that the conditional probability in (A.7) is not affected by the condition $Z > x$, which refers to what happens at or before $t_0 + x$. Therefore, (A.7) becomes

$$P(x+\Delta x) = P(x)\{1 - \rho\Delta x + o(\Delta x)\}, \tag{A.8}$$

and hence one can write

$$P'(x) = -\rho P(x). \tag{A.9}$$

Therefore,

$$P(x) = P(0)e^{-\rho x}, \tag{A.10}$$

and since $P(0) = \text{prob}(Z > 0) = 1$, we have that

$$P(x) = e^{-\rho x}. \tag{A.11}$$

The distribution function of Z is therefore $1 - e^{-\rho x}$, and the probability density function is

$$\rho e^{-\rho x}, \tag{A.12}$$

for $x \geq 0$, which is the exponential distribution with parameter ρ.

Now, consider the process starting from time zero. Let events occur at times $\{Z_1, Z_1 + Z_2, Z_2 + Z_3, ...\}$, i.e., let Z_n be the time interval between the $(n-1)$th and the nth events. If we take $t_0 = 0$ in the argument given above, then Z_1 has the p.d.f. given by (A.12). We can now repeat this argument, taking $t_0 = Z_1$, to show that Z_2 has the p.d.f. given by (A.12) and is independent of Z_1, and so on. Therefore, we can say that the sequence $\{Z_1, Z_2, ...\}$ are independent and identically distributed random variables with the p.d.f. given by (A.12). This property characterises a Poisson process [29].

Also, it follows that if Z has the p.d.f. given by (A.12), then,

$$E(Z) = \frac{1}{\rho}, \tag{A.13}$$

$$V(Z) = \frac{1}{\rho^2}. \tag{A.14}$$

Therefore, since successive intervals are independent, the mean and variance of $Z_1 + ... + Z_r$ are $\frac{r}{\rho}$ and $\frac{r}{\rho^2}$, respectively.

A.3 A Generalisation of the Poisson Process - The Combination of Several Independent Poisson Processes

Suppose that there are k independent Poisson processes of rates $\rho_1, ..., \rho_k$. Let $N(t, t+\Delta t)$ denote the total number of events of any type in $(t, t+\Delta t)$, $N^i(t, t+\Delta t)$ being the number for the ith process. Then,

$$\text{prob}\{N(t, t+\Delta t) = 0\} = \prod_{i=1}^{k} \text{prob}\{N_i(t, t+\Delta t) = 0\} \tag{A.15}$$

$$= \prod_{i=1}^{k} \{1 - \rho_i \Delta t + o(\Delta t)\} \tag{A.16}$$

$$= 1 - \rho\Delta t + o(\Delta t), \tag{A.17}$$

where $\rho = \sum \rho_i$, since the component processes are independent. Similarly,

$$\text{prob}\{N(t, t+\Delta t) = 1\} = \sum_{i=1}^{k} \text{prob}\{N^i(t, t+\Delta t) = 1\} \tag{A.18}$$

$$\text{prob}\{N^j(t, t+\Delta t) = 0; j \neq i\}$$

$$= \rho\Delta t + o(\Delta t). \tag{A.19}$$

Finally, the number of events in $(t, t+\Delta t)$ is independent of occurrences in all processes before or at t. Thus the combined series of events is a Poisson process of rate $\rho = \sum \rho_i$. In particular, if the k separate processes have the same rate ρ_1, the combined process has rate $\rho = k\rho_1$ [29].

Now the time, Z, from the origin to the first event in the combined process has p.d.f. $\rho e^{-\rho x}$. Let $Z_1, ..., Z_k$ be the corresponding times for the separate processes, so that $Z = \min(Z_1, ..., Z_k)$ and $Z_1, ..., Z_k$ are independently distributed with p.d.f.'s $\rho_i e^{-\rho_i x}$ for $(i = 1, ..., k)$. We now consider the joint distribution of Z and of the types

of event to occur first. Given that $Z = x$, the probability that the event is from the first process is,

$$\begin{aligned} \text{prob}\{Z_1 &= x, Z_i > x \ (i = 2, ..., k)|(Z = x\} && \text{(A.20)} \\ &= \frac{\rho_1 e^{-\rho_1 x}.e^{\rho_2 x}...e^{-\rho_k x}}{\rho e^{-\rho x}} && \text{(A.21)} \\ &= \frac{\rho_1}{\rho}. && \text{(A.22)} \end{aligned}$$

It follows that we can describe the set of independent Poisson processes as; intervals between successive events are independently distributed with the p.d.f. $\rho e^{-\rho x}$. Events are then assigned to individual types randomly and with constant probabilities $\frac{\rho_1}{\rho}$, etc.

Appendix B

Implementing the Tree-based Methodology for the Expected Time to Default

Here we briely discuss the compuational implementation of the matrix based solution to the general algorithm for the expected time calculation in Section 4.5.

As one might expect, by representing the default states of firms A, B, C, etc., as a bit string (001...), the size of the probability transition matrix $\underline{\underline{\Omega}}$ quickly becomes very large as the number of firms in the network is increased. In fact, for n_0 firms, the number of matrix elements is 2^{2n_0}. However, this matrix is extremely sparse as shown in Figure B.1 for $n_0 = 10$.

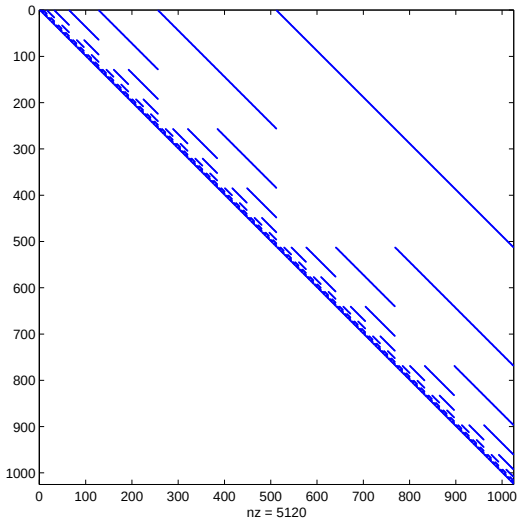


Figure B.1: A spy plot illustrating the sparse matrix structure of a probability transition matrix for $n_0 = 10$. nz is the number of non-zero elements.

Therefore, after some research into the various sparse matrix formats, [30], for a fast implementation we used the built in Matlab sparse matrix. C++ was used to generate the probability transition matrices, $\underline{\underline{\Omega}}$, and the expected time vectors, $\underline{\mathcal{E}}$, in matlab sparse format where required, and then Matlab was used to calculate the matrix products $\underline{\underline{\Omega}}^i$.

We found it possible to work with networks up to a size of $n_0 = 16$, where times for the matrix generation in C++ and matrix products in Matlab were around 250sec and 10sec, respectively.

Appendix C

Explicit Calculations of Expected Default Times for 3 Vertex Networks

Using the tree based methodology, here we calculate explicitly the expected time to default for the 3 vertex 3 edge networks studied in Section 5.1 and shown again in Figure C.1.

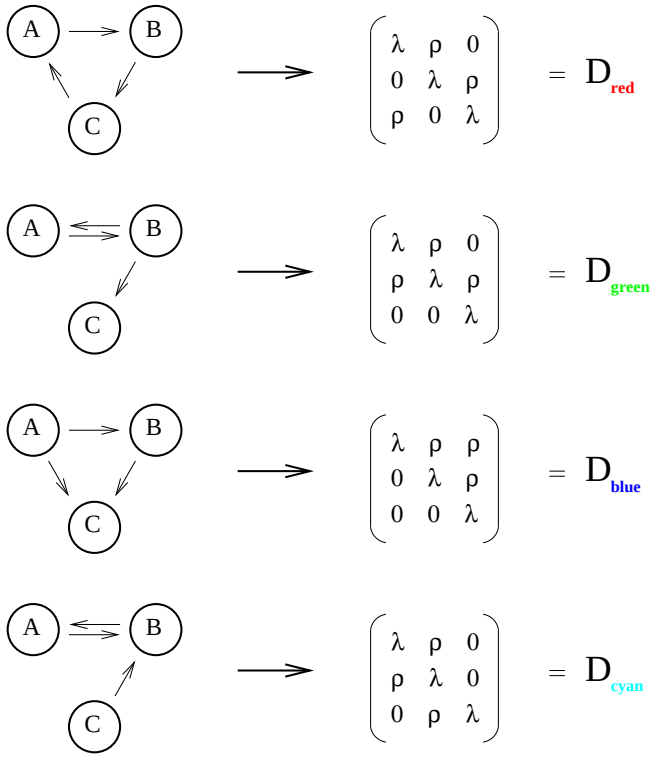


Figure C.1: Default matrices and graph structures for the 4 possible combinations of 3 vertices and 3 edges.

C.1 Expected Default Times for the 3-Vertex, 3-Edge Green Network

Refering to the tree based method for calculating the expected default times, given in Section 4.4, we can write down the tree of probabilities of each order of defaults occuring, as shown in Figure C.2.

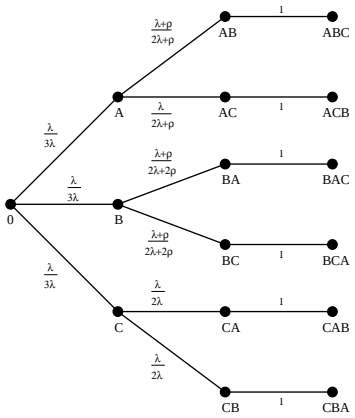


Figure C.2: Probability tree for possible default states of 3-vertex, 3-edge green network.

Therefore, the expected times to default are just calculated from the expected times to be in each of the states (AB, BAC, etc.) multiplied by th associated probabilities of being in those states. Hence, the expected time for 1 default to occur, i.e., the expected time to first be in the state $a = 2$, where a is the number of firms remaining, is given by

$$E(T_2) \;=\; \frac{1}{3\lambda}. \tag{C.1}$$

The expected time for 2 defaults to occur is then just

$$E(T_1) \;=\; E(T_2) + \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda+2\rho)} + \frac{1}{3}\frac{1}{(2\lambda)} \tag{C.2}$$

and the expected time for all 3 firms to default is

$$E(T_0) = E(T_1) \;+\; \frac{1}{3}\Big[\frac{(\lambda+\rho)}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)}\Big] \tag{C.3}$$

$$+\; \frac{1}{3}\Big[\frac{(\lambda+\rho)}{(2\lambda+2\rho)}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{(2\lambda+2\rho)}\frac{1}{(\lambda+\rho)}\Big] \tag{C.4}$$

$$+\; \frac{1}{3}\Big[\frac{\lambda}{2\lambda}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{2\lambda}\frac{1}{(\lambda+\rho)}\Big]. \tag{C.5}$$

C.2 Expected Default Times for the 3-Vertex, 3-Edge Red Network

For the red network, the probability tree is shown in Figure C.3.

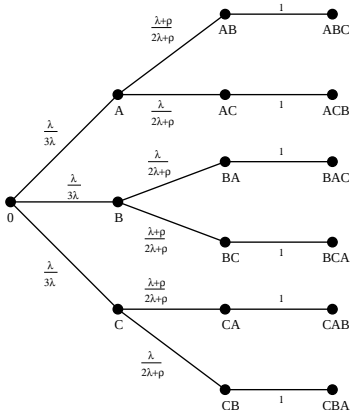


Figure C.3: Probability tree for possible default states of 3-vertex, 3-edge red network.

The corresponding first passage times for 2, 1 and zero firms remaining are

$$E(T_2) = \frac{1}{3\lambda}, \tag{C.6}$$

$$E(T_1) = E(T_2) + \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda+\rho)} \tag{C.7}$$

and

$$E(T_0) = E(T_1) + \frac{1}{3}\Big[\frac{(\lambda+\rho)}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)}\Big] \tag{C.8}$$

$$+ \frac{1}{3}\Big[\frac{(\lambda)}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)}\Big] \tag{C.9}$$

$$+ \frac{1}{3}\Big[\frac{\lambda+\rho}{2\lambda+\rho}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{2\lambda+\rho}\frac{1}{(\lambda+\rho)}\Big]. \tag{C.10}$$

C.3 Expected Default Times for the 3-Vertex, 3-Edge Blue Network

For the blue network, the probability tree is shown in Figure C.4.

The corresponding first passage times for 2, 1 and zero firms remaining are

$$E(T_2) = \frac{1}{3\lambda}, \tag{C.11}$$

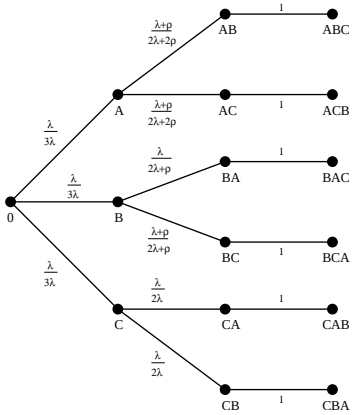


Figure C.4: Probability tree for possible default states of 3-vertex, 3-edge blue network.

$$E(T_1) = E(T_2) + \frac{1}{3}\frac{1}{(2\lambda+2\rho)} + \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda)} \tag{C.12}$$

and

$$E(T_0) = E(T_1) + \frac{1}{3}\Big[\frac{(\lambda+\rho)}{(2\lambda+2\rho)}\frac{1}{(\lambda+2\rho)} + \frac{\lambda+\rho}{(2\lambda+2\rho)}\frac{1}{(\lambda+\rho)}\Big] \tag{C.13}$$

$$+ \frac{1}{3}\Big[\frac{(\lambda)}{(2\lambda+\rho)}\frac{1}{(\lambda+2\rho)} + \frac{\lambda+\rho}{(2\lambda+\rho)}\frac{1}{(\lambda)}\Big] \tag{C.14}$$

$$+ \frac{1}{3}\Big[\frac{\lambda}{2\lambda}\frac{1}{(\lambda+\rho)} + \frac{\lambda}{2\lambda}\frac{1}{(\lambda)}\Big]. \tag{C.15}$$

C.4 Expected Default Times for the 3-Vertex, 3-Edge Cyan Network

And finally, for the cyan network, the probability tree is shown in Figure C.3.

The corresponding first passage times for 2, 1 and zero firms remaining are

$$E(T_2) = \frac{1}{3\lambda}, \tag{C.16}$$

$$E(T_1) = E(T_2) + \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda+\rho)} \tag{C.17}$$

and

$$E(T_0) = E(T_1) + \frac{1}{3}\Big[\frac{(\lambda+\rho)}{(2\lambda+\rho)}\frac{1}{(\lambda)} + \frac{\lambda}{(2\lambda+\rho)}\frac{1}{(\lambda+2\rho)}\Big] \tag{C.18}$$

$$+ \frac{1}{3}\Big[\frac{(\lambda+\rho)}{(2\lambda+\rho)}\frac{1}{(\lambda)} + \frac{\lambda}{(2\lambda+\rho)}\frac{1}{(\lambda+\rho)}\Big] \tag{C.19}$$

$$+ \frac{1}{3}\Big[\frac{\lambda}{2\lambda+\rho}\frac{1}{(\lambda+2\rho)} + \frac{\lambda+\rho}{2\lambda+\rho}\frac{1}{(\lambda+\rho)}\Big]. \tag{C.20}$$

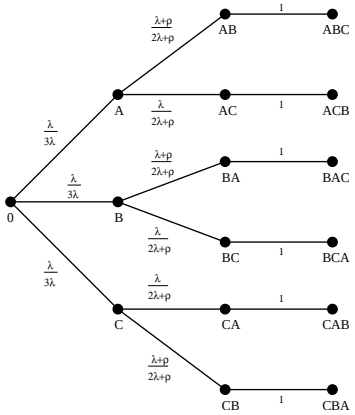


Figure C.5: Probability tree for possible default states of 3-vertex, 3-edge cyan network.

C.5 Relative Position of Expected Times to Default

Once we have calculated the expected times for each order of default for all 4 of the 3-vertex, 3-edge networks, it is straightforward to show that the relative positions of the expected times to default, shown in Figure 5.2, are the same for all values of $\rho > 0$. For example, to show that the expected time for the second default event for the green network is greater than for the red network for all $\rho > 0$ and $\lambda > 0$, we just compare the expected times for the second default events to occur. Therefore, by comparing (C.2) and (C.7) we find that

$$E(T_2)^{Red} \;<\; E(T_2)^{Green}, \tag{C.21}$$

and this is only true when

$$\frac{1}{(2\lambda+\rho)} \;<\; \frac{1}{3}\frac{1}{(2\lambda+\rho)} + \frac{1}{3}\frac{1}{(2\lambda+2\rho)} + \frac{1}{3}\frac{1}{(2\lambda)}. \tag{C.22}$$

Now, by making the substitution

$$x = \frac{\rho}{\lambda}, \tag{C.23}$$

and multiplying through by 2λ we find that

$$\frac{1}{1+x} \;<\; \frac{1}{3}\Big[\frac{1}{1+x} + \frac{1}{1+2x} + 1\Big], \tag{C.24}$$

which gives

$$x^2 > 0 \tag{C.25}$$

and hence

$$\pm\frac{\rho}{2\lambda} > 0. \tag{C.26}$$

Therefore, for all $\lambda > 0$ and $\rho > 0$, the expected time for the second default for the green network is greater than that of the red network.

Appendix D

Derivation of $P_{AB}(t)$ for a General 3 Vertex Network

Here we derive the probability of being in state AB at time t for a general 3 vertex default network.

Consider the general default network illustrated in Figure D.1, which has a default matrix given by

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda_A & \rho_{AB} & \rho_{AC} \\ \rho_{BA} & \lambda_B & \rho_{BC} \\ \rho_{CA} & \rho_{CB} & \lambda_C \end{pmatrix}.$$

The default tree associated with this network is illustrated in Figure D.2 where the path leading to state ABC is highlighted in red. Unlike previous default trees, here on each of the highlighted branches we have labelled the intensity associated with a move between the states.

First considering state 0, we can write down

$$P_0(t+dt) = P_0(t)\Big[(1-\lambda_A dt)(1-\lambda_B dt)(1-\lambda_C dt)\Big]. \tag{D.1}$$

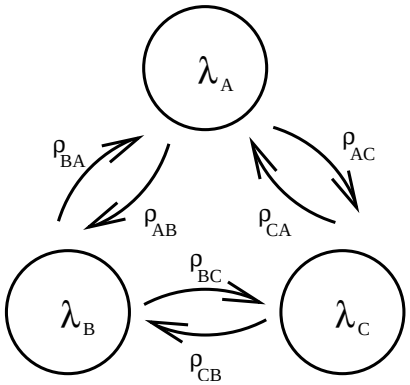


Figure D.1: Default network for 3 reference credits A, B, and C.

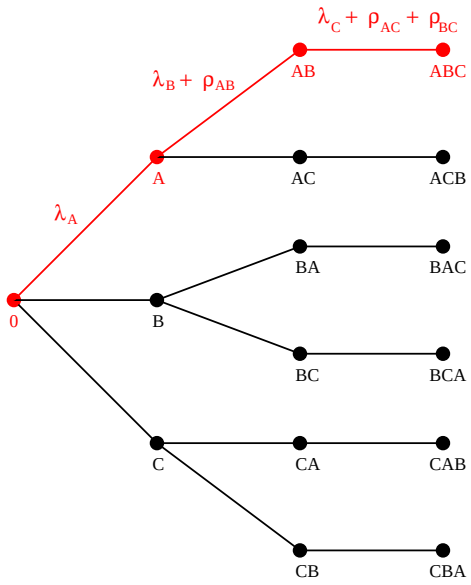


Figure D.2: Default tree for 3 reference credits A, B, and C, with default path leading to state ABC highlighted in red.

Using a Taylor expansion we have that,

$$\frac{dP_0(t)}{dt} = (-\lambda_A - \lambda_B - \lambda_C)P_0(t). \tag{D.2}$$

By applying the initial condition, $P_0(0) = 1$, we have

$$P_0(t) = e^{-(\lambda_A+\lambda_B+\lambda_C)t}. \tag{D.3}$$

Now we have an expression for the probability of being in state 0 at time t.

Let us next work along the highlighted branches in the default tree in Figure D.2, and consider the probability of being in state A at time t. Initially we can write down

$$P_A(t+dt) = P_0(t)((\lambda_A + \lambda_B + \lambda_C)dt.\frac{\lambda_A}{(\lambda_A + \lambda_B + \lambda_C)} \tag{D.4}$$

$$+P_A(t)\Big[1 - (\lambda_B + \rho_{AB}).dt\Big]\Big[1 - (\lambda_C + \rho_{AC})dt\Big]. \tag{D.5}$$

Again, by using a Taylor expansion we have

$$\frac{dP_A(t)}{dt} = \lambda_A P_0(t) - (\lambda_B + \rho_{AB} + \lambda_C + \rho_{AC})P_A(t) \tag{D.6}$$

and if we substitute in our expression for $P_0(t)$ given in (D.3) we arrive at

$$\frac{dP_A(t)}{dt} = \lambda_A e^{-(\lambda_A+\lambda_B+\lambda_C)t} - (\lambda_B + \rho_{AB} + \lambda_C + \rho_{AC})P_A(t), \tag{D.7}$$

which we can solve using the initial condition $P_A(0) = 0$ to get

$$P_A(t) = \frac{(\lambda_A e^{-(\lambda_B+\rho_{AB}+\lambda_C\rho_{AC})t})}{(\rho_{AB}+\rho_{AC}-\lambda_A)}\Big[e^{(\rho_{AB}+\rho_{AC}-\lambda_A)t} - 1\Big]. \tag{D.8}$$

Now we have an expression for $P_A(t)$ we can finally calculate $P_{AB}(t)$. Firstly we can write down

$$P_{AB}(t+dt) = P_A(t)(\lambda_B - \rho_{AB}) + P_{AB}(t)\Big[1 - (\lambda_C + \rho_{AC} + \rho_{BC})\Big]. \tag{D.9}$$

By making a Taylor expansion of $P_{AB}(t+dt)$ we have

$$\frac{dP_{AB}(t)}{dt} = (\lambda_B + \rho_{AB})P_A(t) - (\lambda_C + \rho_{AC} + \rho_{BC})P_{AB}(t). \tag{D.10}$$

Using the initial condition $P_{AB}(0) = 0$, we can solve equation D.10 to get

$$\begin{aligned} P_{AB}(t) &= \frac{\lambda_A(\lambda_B+\rho_{AB})}{(\lambda_A-\rho_{AB}-\rho_{AC})}\Big[\frac{e^{(\lambda_B-\rho_{AC}-\rho_{BC}+\lambda_A)t}}{(\lambda_B-\rho_{AC}-\rho_{BC}+\lambda_A)} - \frac{e^{(-\lambda_B-\rho_{AB}+\rho BC)t}}{(\lambda_B+\rho_{AB}-\rho_{BC})}\Big] \\ &\quad + \Big[\frac{\lambda_A(\lambda_B+\rho_{AB})}{(\lambda_B-\rho_{AC}-\rho_{BC}+\lambda_A)(\lambda_B+\rho_{AB}-\rho_{BC})}\Big]e^{-(\lambda_C+\rho_{AC}+\rho_{BC})t}. \end{aligned} \tag{D.11}$$

Appendix E

Test of Tree Based Algorithm Versus Monte Carlo

In this section we study the accuracy of our tree based approach in calculating the probability of reaching a particular state after a certain amount of time, by comparing it with a Monte Carlo simulation.

Figure E.1 shows the simple default network we are considering which relates to a default matrix given by,

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} \lambda & \rho & 0 \\ 0 & \lambda & \rho \\ 0 & 0 & \lambda \end{pmatrix}.$$

We are interested in the probability of state AB being entered between $t = 0$ and $t = 1000$ for $\lambda = 0.0008$ and $\rho = 0.0016$. Therefore, we must consider the probability of being in state A at time t and then defaulting out of this state into state AB. Firstly, the probability of being in state A at time t is given by,

$$P_A(t) = (e^{\lambda t} - 1)e^{-(2\lambda+\rho)t}. \tag{E.1}$$

Therefore, the probability of entering state AB is just,

$$P_A(t)(\lambda + \rho). \tag{E.2}$$

In Figure E.2 we show a plot of the probability of hitting state AB as a function of time for $\lambda = 0.0008$ and $\rho = 0.0016$.

Hence, the probability of entering state AB between $t = 0$ and $t = 1000$ for $\lambda = 0.0008$ and $\rho = 0.0016$, is,

$$\begin{aligned} \text{Prob } \{ \text{ entering AB between t=0 and t=1000 } \} &= \int_{t=0}^{t=1000} (\lambda + \rho) p_A(t) dt \\ &= 0.1898536997. \end{aligned} \tag{E.3}$$

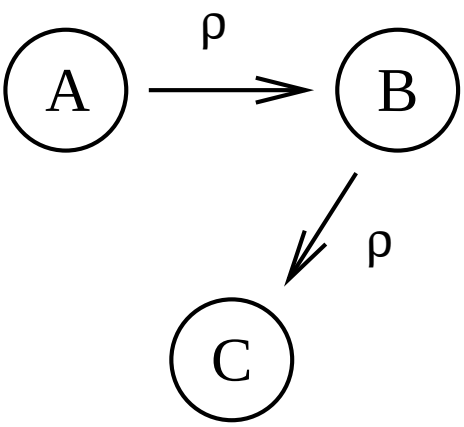


Figure E.1: Default network used in Monte Carlo comparison.

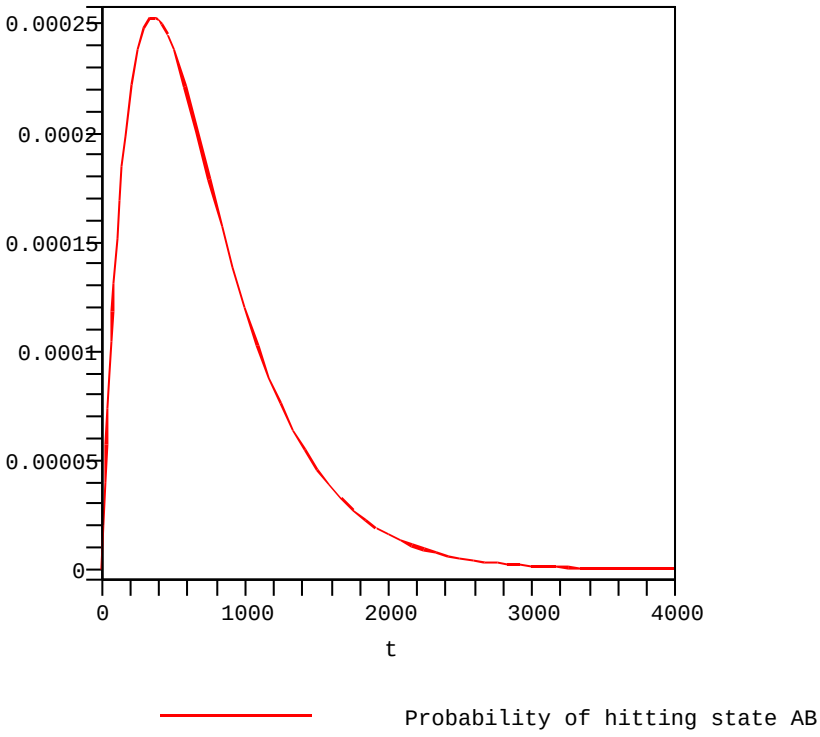


Figure E.2: Plot of the probability of hitting state AB versus time for $\lambda = 0.0008$ and $\rho = 0.0016$.

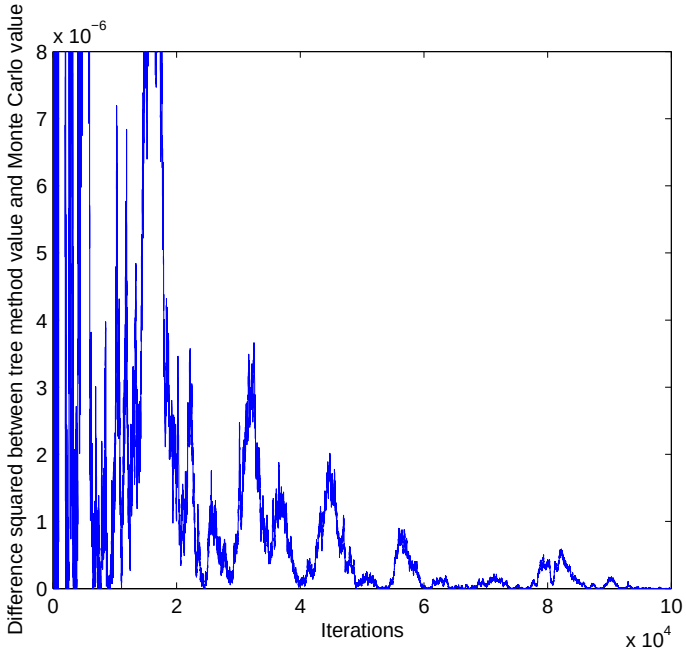


Figure E.3: Plot showing the convergence between the Monte Carlo calculation of the probability of hitting state AB between $t = 0$ and $t = 1000$ for $\lambda = 0.0008$ and $\rho = 0.0016$, and the tree based approach. Value plotted is the squared difference between Monte Carlo and tree method values for increasing number of runs of Monte Carlo simulation.

If we compare this to a Monte Carlo averaging of our default mechanism acting on the network under consideration, over 10 million iterations we get,

$$\text{Prob } \{ \text{ entering AB between t=0 and t=1000 } \} \quad = \quad 0.189715. \tag{E.4}$$

The difference between the value using the Monte Carlo method and the value using the tree-based method is 0.00013 (a difference of less than 0.1%).

One can see, in Figure E.3, the convergence between the Monte Carlo value and the tree based value as we increase the number of Monte Carlo runs.

Although far from a comprehensive test, given the results above, and other comparisons for different networks structures and default parameters, we conclude that our simple Monte Carlo method and the tree-based approach compare well. Therefore, for networks with a large number of vertices, the Monte Carlo approach is a fair approximation to using the significantly more computationally intensive tree-based method.

Appendix F

Symbols and Terms For Agent Based Model of Firm Default

Below are provided some tables of symbols and terminology used in the description and analysis of the agent network model of default.

Table F.1: Agent Network Model.

α	Default growth amount at each iteration (CTD)
$\underline{\underline{A}}$	Adjacency matrix
β	Probability a strategy bit is assessed
$CAAM$	Constant Active Agent Model
CTD	Common Threshold Default Mechanism
δ	Initial default level (CTD)
$D(t)$	Default Level (CTD)
ϵ	Existing cost
g	Score growth period for default assessment (SGD)
g_{start}	Number of iterations before a new agent's score growth is assessed
$\underline{\underline{H}}$	Hamming matrix
κ_{max}, κ_{min}	Maximum and minimum values for $\mu_{i,max}$
λ	Linking cost
$\mu_{i,max}$	Maximum value of each integer strategy bit for each agent
$\mu_{redrawn}$	Probability $\mu_{i,max}$ is redrawn
N	Number of agents
$\underline{\underline{\Psi}}$	Dependence matrix
q	Number of strategy bits or technologies
R_i	Strategy vector
$S_i(t)$	Score of agent i at time t
SGD	Score Growth Default Mechanism
z	Probability a strategy bit is set to zero

Table F.2: New Agent Types.

$best$	Agent copy based on one of the best performing agents
$dead$	Agent copy based on agent just defaulted
g_{size}	Size of group of best performing agents
q_{best}	Probability a strategy bit redrawn for best new agent type
q_{dead}	Probability a strategy bit redrawn for dead new agent type
$r_{rand} : r_{dead} : r_{best}$	Ratios of new agent types generated
$rand$	Random new agent

Table F.3: Small World Analysis of Agent Network Model.

$E(\Gamma_v)$	Set of edges in the neighbourhood of v
γ	Clustering coefficient of whole graph
γ_v	Clustering coefficient of vertex v
Γ_v	Neighbourhood of vertex v - all vertices attached to v
K_v	Number of vertices in Γ_v
l^{-1}	Harmonic mean path length
v	Vertex

Table F.4: Auto-correlation analysis.

i	Iteration index
k	Time lag
N_{it}	Number of iterations
$r(k)$	Auto-correlation
Y_i	Number of defaults at iteration i

Table F.5: Hurst Exponent Analysis.

H	Hurst Exponent
$\mathcal{H}$	Hurst function
τ	Time lag

Appendix G

Additional Graphs for Option Hedger Minority Game

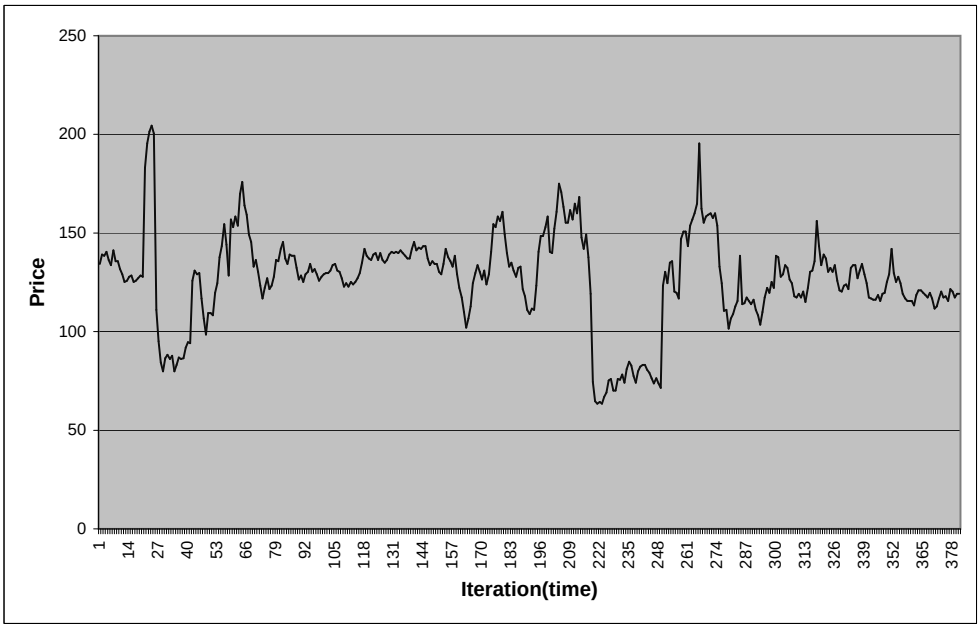


Figure G.1: Change in price with time. From same run as in figure 17.1. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 128.512$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

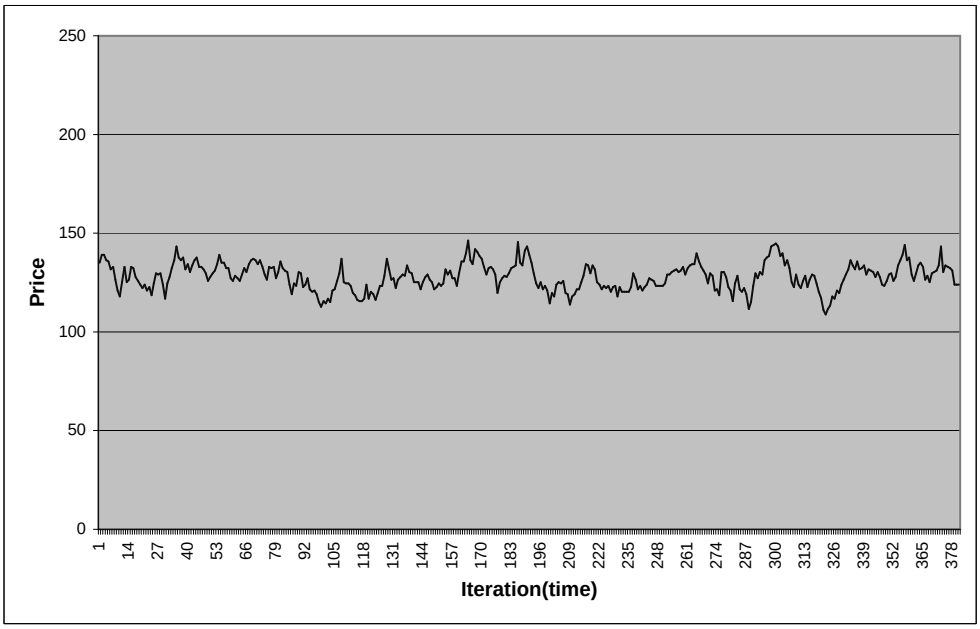


Figure G.2: Change in price with time. From same run as in figure 17.1. Taken from a period with no option hedger acting. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 128.512$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

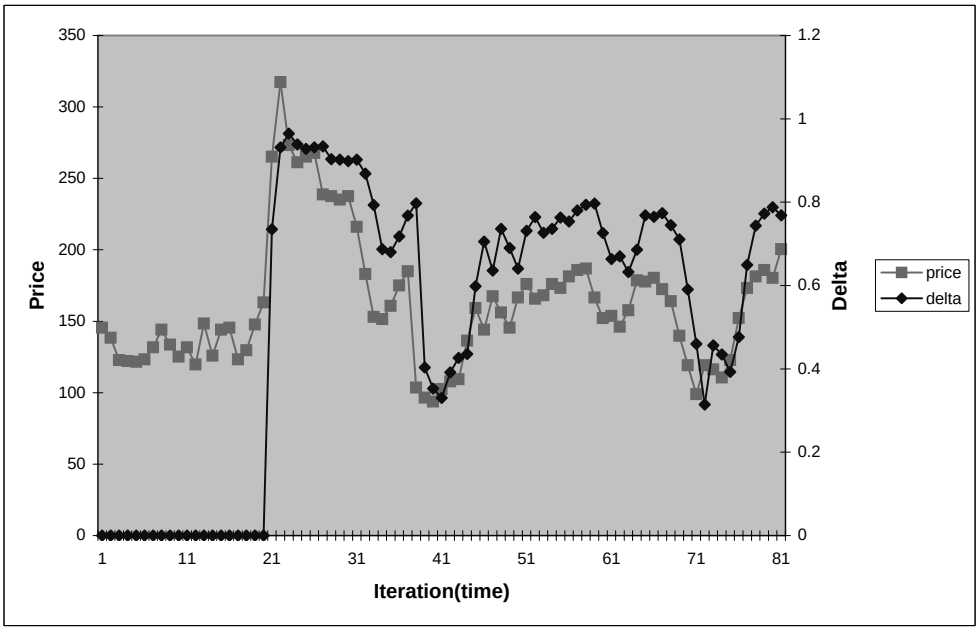


Figure G.3: Change in price and change in delta with time, hedging a vanilla call option. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 163.982$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

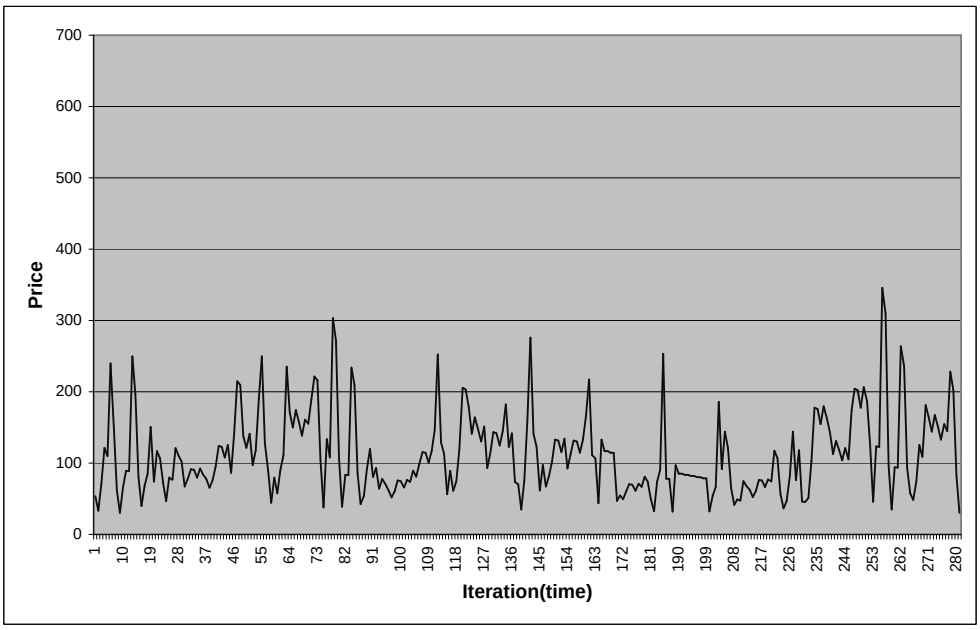


Figure G.4: Change in price with time. From same run as in figure 17.7. Taken from a period with no option hedger acting. $T = 200$, $m = 2$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 151.432$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

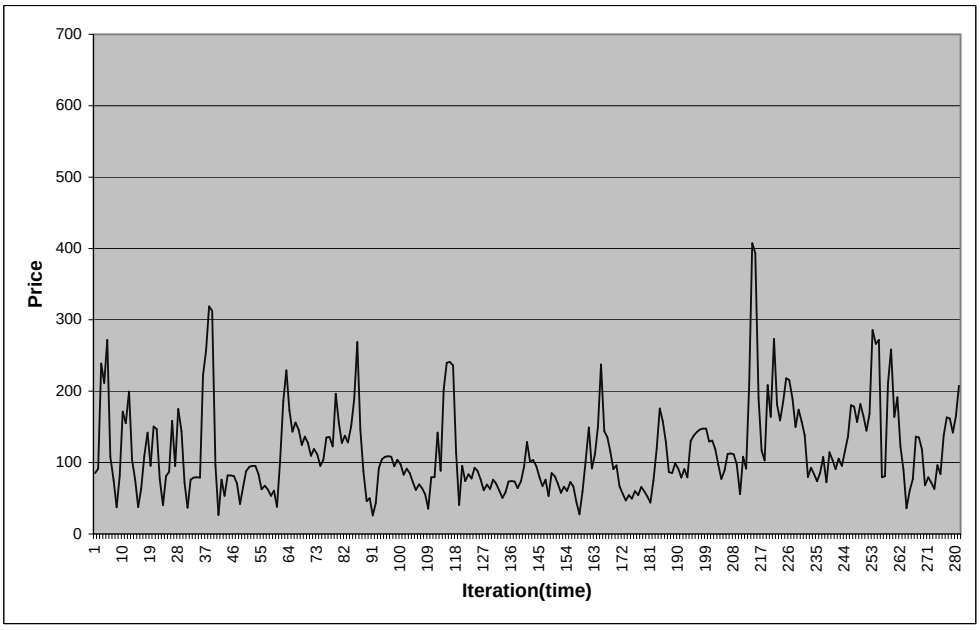


Figure G.5: Change in price with time. From same run as in figure 17.7. Contract starts on iteration 20. $T = 200$, $m = 2$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 151.432$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

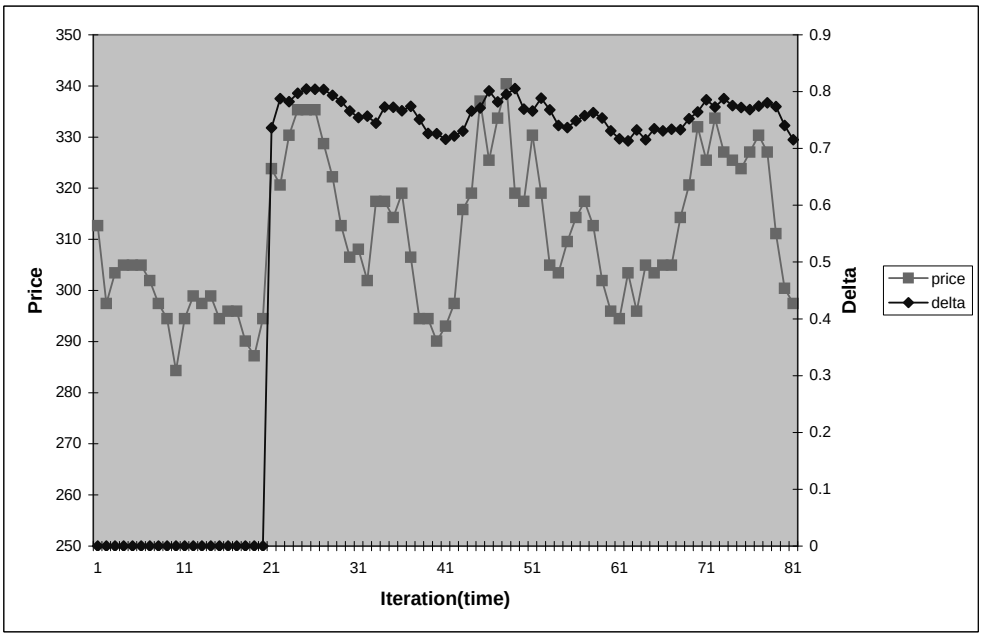


Figure G.6: Change in price and change in delta with time, hedging a vanilla call option. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 295.218$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 25$.

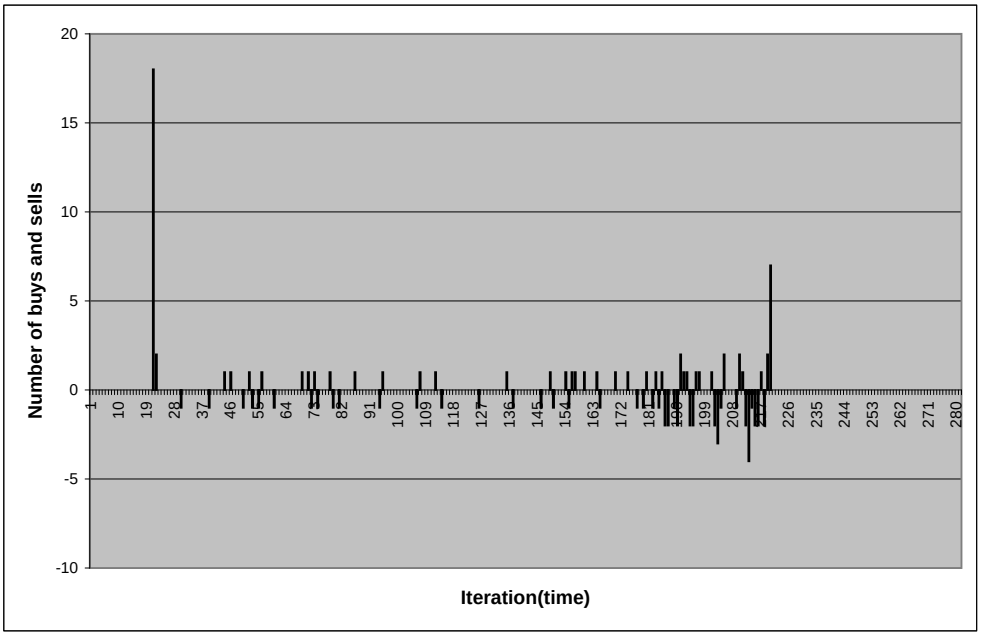


Figure G.7: Number of buys and sells of option hedging agent over life of contract. Positive is buy, negative is sell. From same run as in figure G.6. Contract starts on iteration 20.

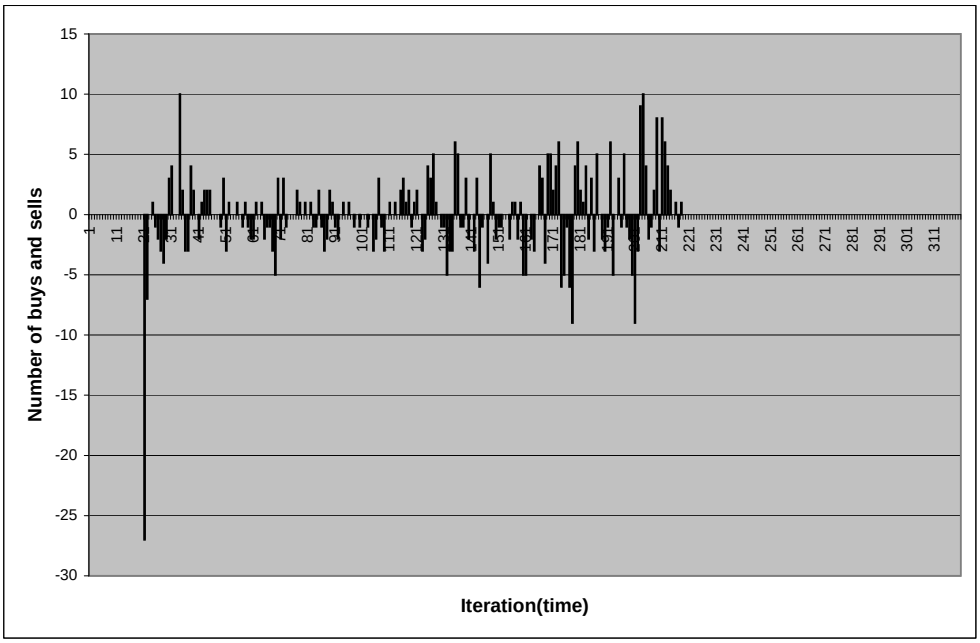


Figure G.8: Number of buys and sells of option hedging agent over life of contract. Positive is buy, negative is sell. From same run as in figure 17.10. Contract starts on iteration 20.$T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 167.279$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

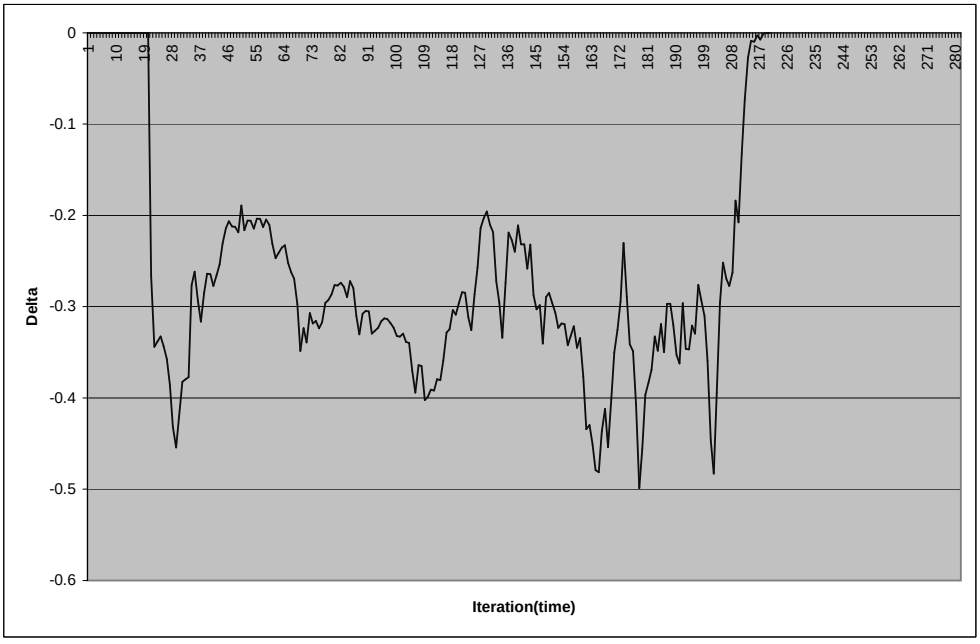


Figure G.9: Change in delta with time. From same run as in figure G.8. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 167.279$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

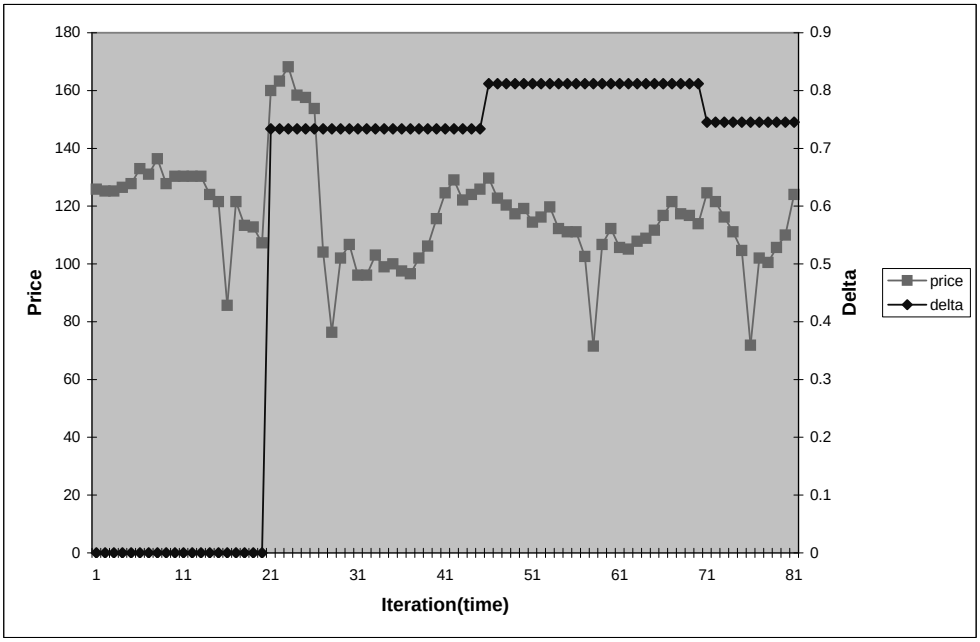


Figure G.10: Change in price and change in delta with time, hedging a vanilla call option. Rehedging every 25th iteration. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 108.001$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

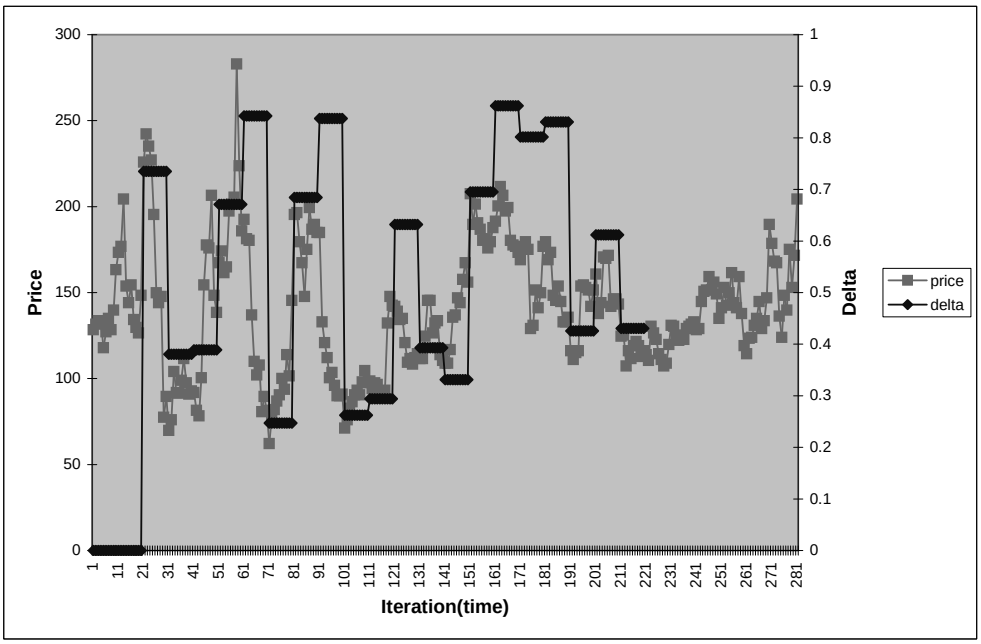


Figure G.11: Change in price and change in delta with time, hedging a vanilla call option. Rehedging every 10th iteration. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 149.188$, $r = 0.001$, $\sigma = 0.04$, $N_{opt} = 100$.

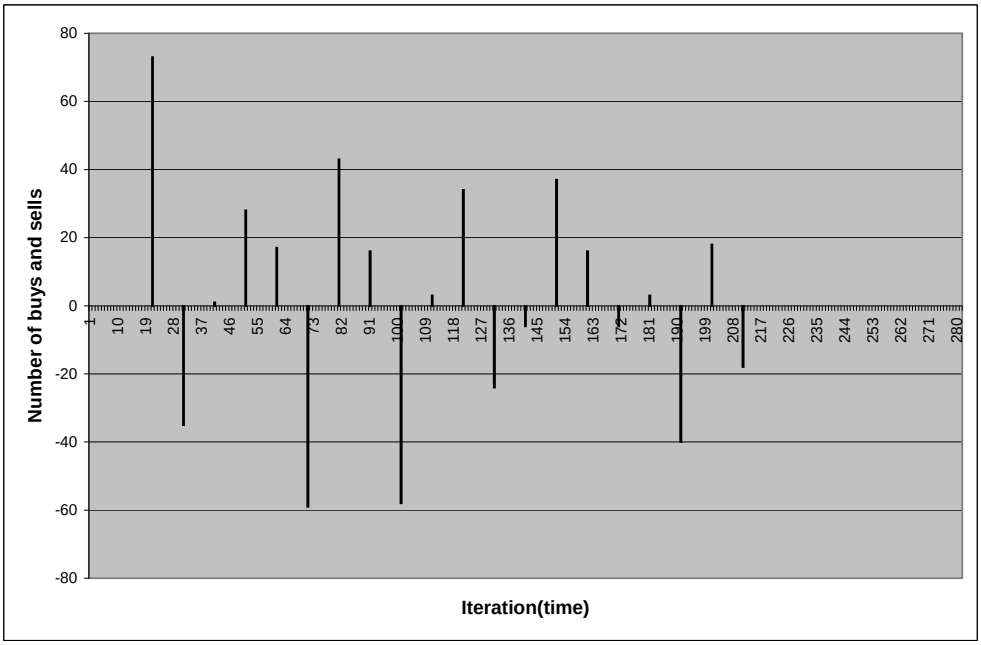


Figure G.12: Number of buys and sells of option hedging agent over life of contract. Positive is buy, negative is sell. From same run as in figure G.11. Contract starts on iteration 20.

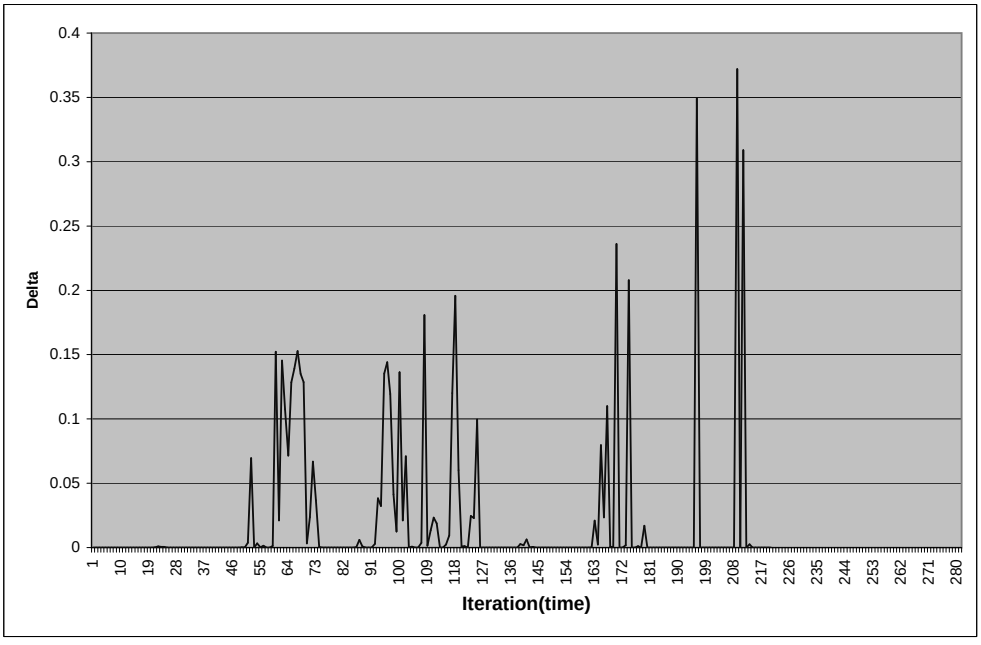


Figure G.13: Change in delta with time. From same run as in figure 17.12. Contract starts on iteration 20.$T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 50.1576$, $r = 0.001$, $\sigma = 0.004$, $N_{opt} = 100$.

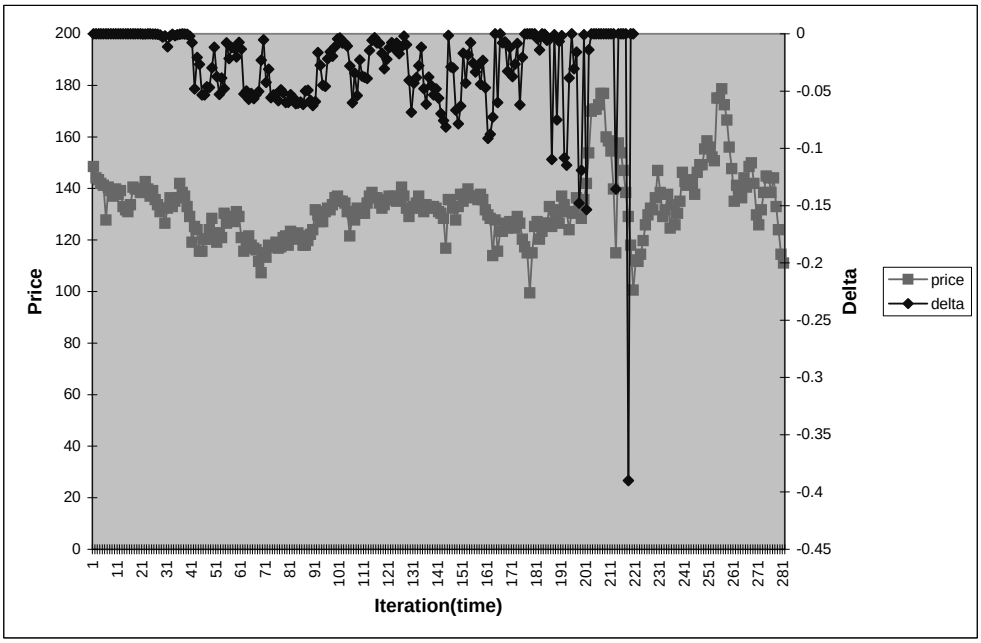


Figure G.14: Change in price and change in delta with time, hedging a binary put option. Contract starts on iteration 20. $T = 200$, $m = 4$, $s = 2$, $N_S = 501$, $N_P = 1000$, $\epsilon = 0.01$, $E = 139.153$, $r = 0.001$, $\sigma = 0.004$, $N_{opt} = 100$.

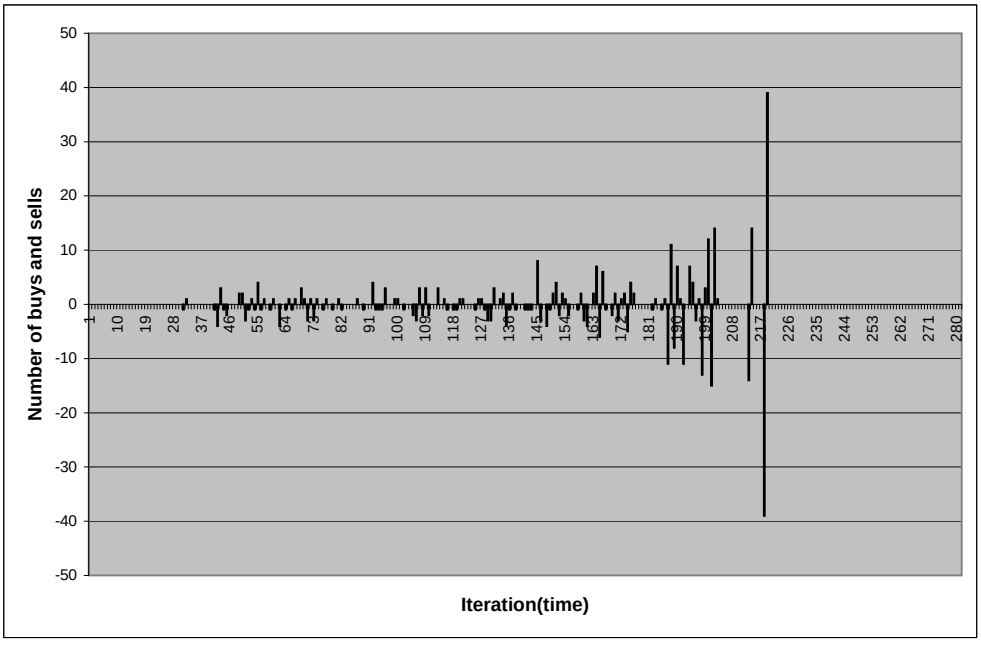


Figure G.15: Number of buys and sells of option hedging agent over life of contract. Positive is buy, negative is sell. From same run as in figure G.14. Contract starts on iteration 20.

References

[1] Dominic O'Kane, Lehman Brothers Credit Derivative Research, 10 February 2004.

[2] John C. Hull, Options, Futures, and Other Derivatives, Prentice-Hall International.

[3] Wolfgang Schmidt, Ian Ward, Pricing Default Baskets, Risk January 2002.

[4] Hans Boscher, Ian Ward, Long or Short in CDOs, Risk June 2002.

[5] Roy Marshal, Marco Naldi, Extreme events and default baskets, Risk June 2002.

[6] Dominic O'Kane, Lutz Schlogl, Modelling Credit: Theory and Practice, Lehman Brothers Analytical Research Series.

[7] Sanjiv R. Das, Laurence Freed, Gary Geng, Nikunji Kapadia, Correlated Default Risk, August 2002.

[8] Fischer Black, Myron Scholes, The pricing of options and corporate liabilities, Journal of Political Economy 81, 81-98, 1973.

[9] Robert C. Merton, On the Pricing of Corporate Debt: The Risk Structure of Interest Rates, Journal of Finance, Vol. 29, MIT (1974), pp. 449-470.

[10] Robert Geske, The Valuation of Corporate Liabilities as Compound Options, Journal of Finance and Quantitative Analysis, November 1977.

[11] Peter Crosbie, Ahmet Kocagil, Modeling Default Risk, Moody's KMV, December 18, 2003.

[12] Kay Giesecke, Credit Risk Modeling and Valuation: An Introduction, August 19, 2002.

[13] Robert A. Jarrow, Stuart M. Turnbull. Pricing Derivatives on Financial Securities Subject to Credit Risk, Journal of Finance, Vol. L, No. 1, Cornell University, and Queen's University (Canada) (Mar-1995), pp. 53-85.

[14] David Lando, On Cox Processes and Credit Risky Securities, March 1998.

[15] Darrell Duffie, Kenneth Singleton, Simulating Correlated Defaults, May 21 1999.

[16] Mark Davis, Violet Lo, Infectious Defaults, Quantitative Finance 1 (2001) 382-387

[17] Daniel Egloff, Marcus Leippold, Paolo Vanini, A Simple Model of Credit Contagion, working paper, February 18, 2004.

[18] Kay Giesecke, Stefan Weber, Cyclical Correlations, Credit Contagion, and Portfolio Losses, November 11, 2003, To appear in Journal of Banking and Finance.

[19] Steven H. Strogatz, Exploring Complex Networks, Nature Vol 410, 8 March 2001.

[20] Brian Hayes, Graph Theory in Practice, American Scientist, Volume 88, Number 1, January-February, 2000, pages 9-13.

[21] Duncan J. Watts, Steven H. Strogatz, Collective dynamics of "small-world" networks, Nature 393, 440 (1998).

[22] Stefano Battiston, Gerrard Weisbuch, Eric Bonabeau, Decision Spread in the Corporate Board Network, Advances in Complex Systems, Vol. 6, No. 4 (2003) 631-644.

[23] Stefano Battiston, Diego Garlaschelli, Guido Caldarelli, The hidden topology of market investment networks, Draft, December 10 2003.

[24] Salvatore Micciche, Giovanni Bonanno, Fabrizio Lillo, Rosario N. Mantegna, Degree stability of a minimum spanning tree of price return and volatility, cond-mat/0212338 v1 14 Dec 2002.

[25] J.P Onnela, A Chakraborti, K. Kaski, J. Kertesz, A. Kanto, Asset trees and asset graphs in financial markets, cond-mat/0303579 v1 27 Mar 2003.

[26] L. Kullmann, J. Kertesz, K. Kaski, Time dependent cross correlations between different stock returns: A directed network of influence, cond-mat/0203256 v2 28 May 2002.

[27] Robert A. Jarrow, Fan Yu, Counterparty Risk and the Pricing of Defaultable Securities, The Journal of Finance, Vol. LVI, No.5, Oct. 2001.

[28] Fan Yu, Correlated Defaults and the Valuation of Defaultable Securities, Working Paper given at the 2nd International Conference on Credit Risk in Montreal, April 2004.

[29] D. R. Cox, H. D. Miller, The Theory of Stochactic Processes, Chapman and Hall 1965.

[30] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in C.

[31] Robin Wilson, Introduction to Graph Theory, Longman, March 1996.

[32] Bela Bollobas, Graph Theory: An Introductory Course, Springer Verlag, September 1979.

[33] James P. Crutchfield, The Calculi of Emergence: Computation, Dynamics, and Induction, Physica D 75 (1994) 11-54. Santa Fe Institute Working Paper 94-03-016.

[34] Albert-Laszlo Barabasi, Reka Albert, Emergence of Scaling in Random Networks, Science vol 286, 15 October 1999.

[35] Emily M. Jin, Michelle Girvan, M. E. J. Newman, Structure of growing social networks, Physical Review E, Volume 64, 046132.

[36] Robert Axtell, The Emergence of Firms in a Population of Agents: Local Increasing Returns, Unstable Nash Equilibria, and Power Law Size Distributions, Center on Social and Economic Dynamics, Working Paper No. 3, June 1999.

[37] Stefan Thurner, Rudolf Hanel, Stefan Plicher, Risk trading, network topology and banking regulation, Quantitative Finance Volume 3 (2003) 306-319.

[38] M. E. J. Newman, A model of mass extinction, Journal of Theoretical Biology, February 12, 1997.

[39] Laurence Smith, Paul Ormerod, Helen Johns, A Model of Agent Extinction: Combining Exogenous Shocks and Networks, Volterra Consulting Ltd. March 2002.

[40] A. De Martino, M. Marsili, I. Pérez Castillo, Typical properties of large random economies with linear activities, cond-mat/0309533 v2 2003.

[41] Yoshi Fujiwara, Zipf Law in Firms Bankruptcy, cond-mat/0310062 v1 2003.

[42] Michel L. Goldstein, Steven A. Morris, Gary G. Yen, Fitting to the Power-Law Distribution, cond-mat/0402322 v1 2004.

[43] Paul Ormerod, Laurence Smith, Power-Law Distribution of Lifespans of Large Firms: Breakdown of Scaling, Volterra Consulting Ltd. 2001.

[44] Robert Axtell, The Emergence of Firms in a Population of Agents: Local Increasing Returns, Unstable Nash Equilibria, and Power Law Size Distributions, Centre on Social and Economic Dynamics, Working Paper No. 3, June 1999.

[45] Pajek - Program for Large Network Analysis http://vlado.fmf.uni-lj.si/pub/networks/pajek/.

[46] Duncan J. Watts, Small Worlds, The Dynamics of Networks between Order and Randomness, Princeton University Press, 1999.

[47] M. E. J. Newman, The Structure and function of complex networks, cond-mat/0303516 v1 March 2003.

[48] S. Milgram, The small world problem, Psychology Today 2, 60-67, 1967.

[49] M. E. J. Newman, Ego-centred networks and the ripple effect - or - Why all your friends are weird, cond-mat/0111070 Nov 2001.

[50] Jean-Philippe Bouchaud, Marc Potters, Theory of Financial Risks, From Statistical Physics to Risk Management, Cambridge University Press, 2001.

[51] R.D. Willmann, G.M. Schutz, and Damien Challet, Exact Hurst Exponent and crossover behaviour in a limit order market model, cond-mat/0206446 v1 June 2002.

[52] W. Brian Arthur, Inductive Reasoning and Bounded Rationality, American Economic Review (Papers and Proceedings), 84,406-411,1994.

[53] D.Challet and Y.C.Zhang, Emergence of Cooperation and Organization in an Evolutionary Game, Physica A 246, 407 1997.

[54] *http://www.unifr.ch/econophysics/minority/*

[55] Neil F.Johnson, Michael Hart, P.M.Hui, Crowd effects and volatility in a competitive market, cond-mat/981127 v1 1998.

[56] M.Hart, P.Jefferies, N.F.Johnson, P.M.Hui, Crowd-anticrowd theory of the minority game, Physica A 298, 537 2001.

[57] M.Marsili, D.Challet, Continuum time limit and stationary states of the Minority Game, cond-mat/0102257 v3 2001.

[58] D.Challet, M.Marsili, Y.C.Zhang, Minority game and stylized facts, Physica A 299, 228 2001.

[59] M.Marsili, Market mechanism and expectations in minority and majority games, Physica A 299, 93 2001.

[60] P.Wilmott, Paul Wilmott Introduces Quantitative Finance, Wiley.

[61] J.C.Hull, Options, Futures, and Other Derivatives, Fourth Edition, Prentice Hall.

[62] Black, F. and Scholes, M, The pricing of options and corporate liabilities. J. Pol. Econ. 81 637-659, 1973.

[63] Merton, R.C. Theory of rational option pricing. Bell J. Econ. Manag. Sci. 4, 1973.

[64] D.Challet, M.Marsili, R.Zecchina, Phase Transition in a Toy market from Minority Game website [54].

[65] R.Savit, R.Manuca, R.Riolo, Adaptive Competition, Market Efficiency, and Phase Transitions, Physical Review Letters, Volume 82, Number 10, 1999.

[66] D.Challet, M.Marsili, R.Zecchina, Exact Solution of a Modified El Farol's bar problem: Efficiency and the role of market impact, cond-mat/9908480 v3 1999.

[67] N.F.Johnson, P.Jefferies, P.M.Hui, Physics and Finance, Draft of course notes, Physics Dept. Oxford University, 2002.

[68] D.Challet, M.Marsili, Y.C.Zhang, Stylized facts of financial markets and market crashes in Minority Games, cond-mat/0101326 v1 2001.

[69] D.Challet, M.Marsili, Stylized Facts, finite size effects and control parameters in minority games, pre-print 2002.

[70] M.L.Hart, D.Lamper, N.F.Johnson, Crash Avoidance in a Complex System, cond-mat/0206228 v1 2002.

[71] P.Wilmott, Paul Wilmott on Quantitative Finance, Wiley.

[72] M.A.R. de Cara, O.Pla, F.Guinea, Competition, efficiency and collective behaviour in the "El Farol" bar model, Eur. Phys. J. B 10, 187-191 1999.